

TUGAS AKHIR - KS 141501

PERAMALAN JUMLAH PRODUKSI IKAN DENGAN
MENGUNAKAN *BACKPROPAGATION NEURAL
NETWORK* (STUDI AKSUS: UPTD PELABUHAN
PERIKANAN BANJARMASIN)

*PRODUCTION FORECASTING USING
BACKPROPAGATION NEURAL NETWORK (CASE
STUDY: UPTD PELABUHAN PERIKANAN
BANJARMASIN)*

MUHAMMAD AZHAR RAZAK
NRP 5212 100 138

Dosen Pembimbing:
Edwin Riksakomara, S. Kom, M. T.



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS 141501

**PERAMALAN JUMLAH PRODUKSI IKAN DENGAN
MENGUNAKAN *BACKPROPAGATION NEURAL*
NETWORK (STUDI KASUS: UPTD PELABUHAN
PERIKANAN BANJARMASIN)**

MUHAMMAD AZHAR RAZAK
NRP 5212 100 138

Dosen Pembimbing:
Edwin Riksakomara, S. Kom, M. T.

JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS 141501

***PRODUCTION FORECASTING USING
BACKPROPAGATION NEURAL NETWORK (CASE STUDY:
UPTD PELABUHAN PERIKANAN BANJARMASIN)***

MUHAMMAD AZHAR RAZAK

NRP 5212 100 138

Supervisor:

Edwin Riksakomara, S. Kom, M. T.

JURUSAN SISTEM INFORMASI

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya 2017

LEMBAR PENGESAHAN

PERAMALAN PRODUKSI IKAN DENGAN MENGUNAKAN *BACKPROPAGATION NEURAL NETWORK* (STUDI KASUS: UPTD PELABUHAN PERIKANAN BANJARMASIN)

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

MUHAMMAD AZHAR RAZAK

NRP. 5212 100 138

Surabaya, 2017

KETUA JURUSAN SISTEM INFORMASI



Dr. Ir. Aris Tjahyanto, M. Kom.

NIP 19650310 199102 1 001

LEMBAR PERSETUJUAN

PERAMALAN PRODUKSI IKAN DENGAN MENGUNAKAN *BACKPROPAGATION NEURAL NETWORK* (STUDI KASUS: UPTD PELABUHAN PERIKANAN BANJARMASIN)

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

MUHAMMAD AZHAR RAZAK

NRP. 5212 100 0138

Disetujui Tim Penguji :
Tanggal Ujian : Januari 2017
Periode Wisuda : 115

Edwin Riksakomara, S. Kom., M. T.

(Pembimbing I)

Wiwik Anggraeni, S. Si. M. Kom

(Penguji I)

Faisal Mahananto, S. Kom., M. Eng., Ph. D.

(Penguji II)

PERAMALAN JUMLAH PRODUKSI IKAN DENGAN MENGGUNAKAN *BACKPROPAGATION* *NEURAL NETWORK* (STUDI KASUS: UPTD PELABUHAN PERIKANAN BANJARMASIN)

Nama Mahasiswa : Muhammad Azhar Razak
NRP : 5212100138
Jurusan : Sistem Informasi FTIf - ITS
Pembimbing 1 : Edwin Riksakomara, S. Kom, M. T.

ABSTRAK

Pelabuhan Perikanan Banjarmasin (UPTD dibawah binaan Dinas Perikanan dan Kelautan Kalimantan Selatan) mempunyai salah satu tugas pokok dan fungsi untuk memproduksi hasil tangkapan ikan dari kapal-kapal nelayan dan mendistribusikannya ke berbagai daerah untuk dipasarkan dalam setiap bulannya. Hasil pemasaran ini nantinya akan digunakan Pemerintah Daerah untuk meningkatkan Pendapatan Asli Daerah (PAD) dari provinsi itu sendiri. Permasalahan yang ada adalah jumlah produksi ikan ini umumnya tidak menentu pada setiap bulannya yang mungkin juga dapat berdampak pada kegiatan utama lainnya, seperti penyaluran es pendingin, penyediaan air bersih, dan kegiatan lainnya yang berkaitan dengan proses produksi. Hal seperti ini merupakan permasalahan lumrah dalam bidang peramalan, yaitu ketidakpastian suatu keadaan apabila dilihat fakta-fakta history yang telah lalu. Oleh karena itu, pada penelitian ini dicoba untuk mencari solusi dengan penggunaan metode peramalan ANN untuk menganalisa sistemasi kegiatan dalam proses produksi ikan instansi ini.

Artificial Neural Network (ANN) merupakan sebuah sistem pemrosesan data dengan meniru cara kerja sistem saraf manusia. ANN merupakan sebuah sistem yang terdiri atas banyak elemen pemrosesan sederhana yang terhubung secara

paralel. Backpropagation Neural Network (BPNN) dikatakan memiliki kelebihan dalam aspek pembelajaran sistem (adaptive) dan memiliki resiko kesalahan kecil (fault tolerance) terhadap pemecahan masalah.

Didapatkan model peramalan optimal dari hasil penelitian ini dengan masukan jumlah produksi ikan satu bulan sebelumnya dan dua bulan sebelumnya, serta hasil peramalan untuk periode September 2016 adalah 865753 kg. Variabel suhu udara dinilai kurang mempengaruhi terhadap hasil peramalan menggunakan dataset produksi ikan ini dilihat dari error yang mendekati 21%. Keluaran aplikasi peramalan sederhana memiliki fungsi menampilkan history, proses training dan testing, dan menampilkan grafik.

Keywords: UPTD Pelabuhan Perikanan Banjarmasin, Production Forecasting, Backpropagation Neural Network

***PRODUCTION FORECASTING USING
BACKPROPAGATION NEURAL NETWORK (CASE
STUDY: UPTD PELABUHAN PERIKANAN
BANJARMASIN)***

Student Name : Muhammad Azhar Razak
NRP : 5212100138
Department : Sistem Informasi FTIf - ITS
Supervisor 1 : Edwin Riksakomara, S. Kom, M. T.

ABSTRACT

Pelabuhan Perikanan Banjarmasin (UPTD under the guidance of the Dinas Perikanan dan Kelautan Kalimantan Selatan) has one of it's main tasks and functions to produce the catches from fishing vessels and distribute them to the various areas to be marketed in each month. Marketing results will be used to Local Government to increase revenue of the province itself. The problems that exist are fish production is generally erratic on a monthly basis which may also be an impact on other major events, such as the distribution of ice-cooling, water supply, and other activities related to the production process. Things like this is a common problem in the field of forecasting, the uncertainty of a condition when seen facts different from the data history. Therefore, the authors tried to find a solution with the use of ANN forecasting methods to view the systematization of production activities at this institution.

Artificial Neural Network (ANN) is a data processing system by mimicking the workings of the human nervous system. ANN is a system composed of many simple processing elements are connected in parallel. Backpropagation Neural Network (BPNN) is said to have an advantage in learning aspect of the system (adaptive) and have a small risk of error (fault tolerance) toward solving the problem.

Optimal forecasting model obtained from the results of this study with fish production inputs one and two months earlier and forecasting results for the period of September 2016 was 865753 kg. Variabel air temperature is considered less supportive of forecasting using this dataset is seen from error approaching 21%. Simple forecasting application has a function display the history, the process of training and testing, and displaying graphics.

Keywords: UPTD Pelabuhan Perikanan Banjarmasin, Production Forecasting, Backpropagation Neural Network

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas selesainya Tugas Akhir yang berjudul:

**“PERAMALAN JUMLAH PRODUKSI IKAN DENGAN
MENGUNAKAN *BACKPROPAGATION NEURAL
NETWORK* (STUDI KASUS: UPTD PELABUHAN
PERIKANAN BANJARMASIN)”**

Atas dukungan moral dan materil yang diberikan dalam penyusunan Tugas Akhir ini, maka penulis mengucapkan banyak terima kasih kepada

- Allah SWT yang telah memberikan kemudahan dan kelancaran dalam penyelesaian Tugas Akhir ini.
- Keluarga yang selalu senantiasa mendoakan dan memberikan dukungan.
- Bapak Dr. Ir. Aris Tjahyanto, M. Kom., selaku Ketua Jurusan Sistem Informasi ITS, yang telah menyediakan fasilitas terbaik.
- Bapak Edwin Riksakomara, S. Kom., M. T., selaku dosen pembimbing yang telah banyak meluangkan waktu untuk membimbing dan mengarahkan penulis.
- Ibu Wiwik Anggraeni, S. Si., M. Kom., yang telah memberikan materi dengan sangat baik terkait dengan topik yang diangkat.
- Bapak Faisal Mahananto, S. Kom., M. Eng., Ph. D. dan Ibu Amalia Utamima, S. Kom., MBA. yang memberikan arahan dengan baik terkait kekurangan pada laporan Tugas Akhir ini.
- Bapak Arif Wibisono, S. Kom. M. Sc., selaku dosen wali yang senantiasa memberikan arahan selama masa perkuliahan.
- Seluruh dosen pengajar beserta staff dan karyawan di Jurusan Sistem Informasi, FTIf - ITS Surabaya.

- Serta teman-teman civitas akademika yang senantiasa memberikan dukungan moril selama pengerjaan Tugas Akhir ini yang tidak mungkin dapat disebutkan satu-satu.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan di dalamnya. Oleh karena itu, penulis meminta maaf atas segala kesalahan yang dibuat dalam penulisan Tugas Akhir ini. Semoga hasil dari Tugas Akhir ini dapat bermanfaat bagi seluruh pihak. Amin.

Surabaya, 2017

Penulis

DAFTAR ISI

ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xviii
DAFTAR TABEL	xx
DAFTAR <i>SCRIPT</i>	xxi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan Tugas Akhir	3
1.5. Manfaat Tugas Akhir	3
1.6. Relevansi Tugas Akhir	3
BAB II TINJAUAN PUSTAKA	5
2.1. Studi Sebelumnya	5
2.2. Dasar Teori	13
2.2.1. Dinas Perikanan dan Kelautan Provinsi Kalimantan Selatan	13
2.2.2. UPTD Pelabuhan Perikanan Pantai Banjarmasin	14
2.2.3. Variabel Pendukung (Suhu Udara)	15
2.2.3.1. Pengertian Suhu Udara	15

2.2.3.2. Pengaruh Suhu Udara terhadap Habitat Ikan	16
2.2.4. <i>Artificial Neural Network</i> (ANN)	16
2.2.4.1. Algoritma Backpropagation	19
2.2.4.2. Fungsi Aktivasi BPNN	24
2.2.4.2. Parameter BPNN	27
2.2.5. Performa Peramalan	28
BAB III METODOLOGI PENELITIAN	31
3.1. Tahapan Pelaksanaan Tugas Akhir	31
3.1.1. Studi Literatur	32
3.1.2. Pengambilan Data	32
3.1.3. Pembuatan Model	32
3.1.4. Pengujian Model	32
3.1.5. Implementasi Model Optimal	32
3.1.6. Penarikan Kesimpulan	33
3.1.7. Penyelesaian Laporan dan Paper	33
BAB IV PERANCANGAN	35
4.1. <i>Data Input</i>	35
4.2. Model ANN (<i>Artificial Neural Network</i>)	36
4.2.1. Model 1	37
4.2.2. Model 2	37
4.2.3. Model 3	38
4.2.4. Model 4	38
4.2.5. Model 5	39
4.2.6. Model 6	40

4.2.7. Model 7	41
4.2.8. Model 8	42
4.2.9 Model 9	43
4.2.10. Model 10	44
4.3. Perancangan Aplikasi	45
4.3.1. Kebutuhan Fungsional	45
4.3.1.1. Fungsi Menampilkan Data History	45
4.3.1.2. Fungsi Training	46
4.3.1.3. Fungsi Testing	46
4.3.1.4. Fungsi Menampilkan Testing Error	46
4.3.1.5. Fungsi Forecast.....	46
4.3.1.6. Fungsi Menampilkan Grafik.....	46
4.3.2. <i>Use Case Description</i>	47
4.3.2.1. <i>Use Case</i> Menampilkan Data History.....	47
4.3.2.2. <i>Use Case</i> Training Process	48
4.3.2.3. <i>Use Case</i> Testing Process	49
4.3.2.4. <i>Use Case</i> Testing Error.....	50
4.3.2.5. <i>Use Case</i> Forecast	51
4.3.2.6. <i>Use Case</i> Menampilkan Grafik.....	52
BAB V IMPLEMENTASI	55
5.1. Pemasukan Data	55

5.2. Pembentukan Model.....	55
5.2.1. Proses <i>Training</i>	57
5.2.2. Proses <i>Testing</i>	60
5.3. <i>Forecast</i>	61
5.4. Implementasi Aplikasi.....	62
5.4.1. Fungsi Menampilkan Data History	62
5.4.2. Fungsi Training	63
5.4.3. Fungsi Testing	64
5.4.4. Fungsi Menampilkan Testing Error	64
5.4.5. Fungsi Forecast.....	65
5.4.6. Fungsi Menampilkan Grafik.....	66
BAB VI HASIL DAN PEMBAHASAN	69
6.1. Hasil Implementasi Model	69
6.1.1. Hasil <i>Train & Test Error</i> (<i>n hidden layer</i>).....	69
6.1.2. Hasil <i>Train & Test Error</i> (<i>2n hidden layer</i>).....	72
6.1.3. Hasil <i>Train & Test Error</i> (<i>3n hidden layer</i>).....	75
6.2. Analisa Hasil Implementasi.....	78
6.2.1. Analisa Model	78
6.2.2. Analisa Variabel Pendukung (Suhu Udara)	80
6.2.3. Analisa Parameter.....	81
6.2.3.1. <i>Learning Rate</i>	81
6.2.3.2. <i>Epoch</i>	84
6.2.4. Analisa Rasio <i>Dataset</i>	85
6.3. Hasil <i>Forecast</i>	86

6.4. <i>Graphical User Interface</i>	87
6.4.1. Fungsi Menampilkan Data History	88
6.4.2. Fungsi Training	88
6.4.3. Fungsi Testing	89
6.4.4. Fungsi Menampilkan Testing Error	90
6.4.5. Fungsi Forecast.....	91
6.4.6. Fungsi Menampilkan Grafik.....	92
BAB VII KESIMPULAN DAN SARAN	95
7.1. Kesimpulan	95
7.2. Saran	96
DAFTAR PUSTAKA	99
BIODATA PENULIS	103
LAMPIRAN A.....	A-1
LAMPIRAN B	B-1
LAMPIRAN C	C-1
LAMPIRAN D.....	D-1

DAFTAR GAMBAR

Gambar 2. 1. Jaringan Saraf Manusia.....	17
Gambar 2. 2. Arsitektur Multi Layer ANN [14]	18
Gambar 2. 3. Arsitektur ANN Backpropagation	19
Gambar 2. 4. Fungsi <i>Binary Sigmoid</i>	25
Gambar 2. 5. Fungsi <i>Bipolar Sigmoid</i>	25
Gambar 2. 6. Fungsi Bipolar Treshold	26
Gambar 2. 7 Fungsi Linear	26
Gambar 2. 8. <i>Gradient Descent</i>	27
Gambar 3. 1. Metodologi Penelitian.....	31
Gambar 4. 1. Pola Data Variabel Produksi Ikan	35
Gambar 4. 2. Pola Data Variabel Suhu Udara	36
Gambar 4. 3. Model ANN 1	37
Gambar 4. 4. Model ANN 2	37
Gambar 4. 5. Model ANN 3	38
Gambar 4. 6. Model ANN 4	39
Gambar 4. 7. Model ANN 5	40
Gambar 4. 8. Model ANN 6	41
Gambar 4. 9. Model ANN 7	42
Gambar 4. 10. Model ANN 8	43
Gambar 4. 11. Model ANN 9	44
Gambar 4. 12. Model ANN 10	45
Gambar 5. 1. Fungsi Aktivasi <i>Tansig</i>	56
Gambar 5. 2. Fungsi Aktivasi <i>Purelin</i>	56
Gambar 5. 3. Contoh Pembuatan Model	57

Gambar 5. 4. Contoh Proses <i>Training</i>	59
Gambar 5. 5. Contoh Hasil <i>Training</i>	60
Gambar 5. 6. Contoh Proses <i>Testing</i>	61
Gambar 6. 1. Grafik Perbandingan <i>Error (n hidden layer)</i>	72
Gambar 6. 2. Grafik Perbandingan <i>Error (2n hidden layer)</i> ..	74
Gambar 6. 3 Grafik Perbandingan <i>Error (3n hidden layer)</i> ...	77
Gambar 6. 4. Model Optimal.....	78
Gambar 6. 5. Hasil <i>Training</i> Model 3 (<i>n hidden layer</i>)	79
Gambar 6. 6. Hasil <i>Testing</i> Model 3 (<i>n hidden layer</i>)	80
Gambar 6. 7. Hasil <i>Forecast</i> menggunakan Model 3	87
Gambar 6. 8. Jendela Tampilan <i>Data History</i>	88
Gambar 6. 9. Jendela Fungsi <i>Training</i>	89
Gambar 6. 10. Jendela Fungsi <i>Testing</i>	90
Gambar 6. 11. Jendela Tampilan <i>Testing Error</i>	90
Gambar 6. 12. Jendela Fungsi <i>Forecast</i>	91
Gambar 6. 13. Jendela Tabel Hasil <i>Forecast</i>	92
Gambar 6. 14. Fungsi Grafik <i>Data History</i>	92
Gambar 6. 15. Fungsi Grafik Hasil <i>Forecast</i>	93
Gambar 6. 16. Grafik <i>Data History</i>	93

DAFTAR TABEL

Tabel 2. 1. <i>Review Paper 1</i>	6
Tabel 2. 2. <i>Review Paper 2</i>	7
Tabel 2. 3. <i>Review Paper 3</i>	8
Tabel 2. 4. <i>Review Paper 4</i>	9
Tabel 2. 5. <i>Review Paper 5</i>	11
Tabel 2. 6. <i>Review Paper 6</i>	12
Tabel 2. 7. Skala Pengukuran Kinerja Model [16]	29
Tabel 2. 8. Skala Pengukuran Kinerja Variabel.....	29
Tabel 4. 1. <i>Use Case</i> Menampilkan Data History.....	47
Tabel 4. 2. <i>Use Case</i> Training Process	48
Tabel 4. 3. <i>Use Case</i> Testing Process.....	49
Tabel 4. 4. <i>Use Case</i> Testing Error	50
Tabel 4. 5. <i>Use Case</i> Forecast	51
Tabel 4. 6. <i>Use Case</i> Menampilkan Grafik	52
Tabel 6. 1. Tabel Perbandingan <i>Error (n hidden layer)</i>	71
Tabel 6. 2. Tabel Perbanding <i>Error (2n hidden layer)</i>	74
Tabel 6. 3. Tabel Perbandingan <i>Error (3n hidden layer)</i>	76
Tabel 6. 4. Performa <i>Input</i> Variabel Pendukung	81
Tabel 6. 5. Performa Variasi <i>Learning Rate</i>	82
Tabel 6. 6. Performa Variasi <i>Epoch</i>	84
Tabel 6. 7. Performa Variasi Rasio <i>Dataset</i>	86

DAFTAR SCRIPT

<i>Script 5. 1. Fungsi Menampilkan Data History</i>	<i>62</i>
<i>Script 5. 2. Fungsi Training.....</i>	<i>63</i>
<i>Script 5. 3. Fungsi Testing</i>	<i>64</i>
<i>Script 5. 4. Fungsi Menampilkan Testing Error</i>	<i>65</i>
<i>Script 5. 5. Script 1 Fungsi Forecast.....</i>	<i>65</i>
<i>Script 5. 6. Script 2 Fungsi Forecast.....</i>	<i>66</i>
<i>Script 5. 7. Script 1 Fungsi Menampilkan Grafik.....</i>	<i>66</i>
<i>Script 5. 8. Script 2 Fungsi Menampilkan Grafik.....</i>	<i>67</i>

“Halaman ini sengaja dikosongkan”

BAB I

PENDAHULUAN

Pada bab ini akan dijelaskan mengenai proses identifikasi masalah penelitian yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan tugas akhir, manfaat tugas akhir, dan relevansi tugas akhir. Berdasarkan penjelasan pada bab ini, diharapkan pembaca dapat memahami sudut pandang penulis dan permasalahan yang ingin dipecahkan.

1.1. Latar Belakang

Pada masa sekarang ini, peramalan merupakan hal yang lazim digunakan pada dunia perindustrian dan juga pada lembaga pemerintahan. Seperti contohnya adalah peramalan permintaan produksi kain, peramalan fluktuasi saham, peramalan pemakaian internet pada suatu daerah, dan peramalan penjualan produk. Hal ini juga berlaku oleh lembaga pemerintahan, seperti peramalan penerimaan pajak suatu daerah, peramalan kecelakaan lalu lintas, peramalan kepadatan arus mudik, peramalan penyebaran suatu penyakit, dan lain sebagainya.

Peramalan merupakan seni ilmu untuk memprediksi peristiwa-peristiwa masa depan dengan menggunakan data terdahulu dan memproyeksikannya ke masa depan dengan menggunakan pendekatan-pendekatan matematis [1].

Pelabuhan Perikanan yang merupakan salah satu UPTD di bawah binaan Dinas Perikanan dan Kelautan Provinsi Kalimantan Selatan memiliki salah satu tugas pokok untuk memproduksi hasil tangkapan ikan dan mendistribusikannya ke berbagai daerah untuk dipasarkan. Hasil pemasaran dari distribusi dan produksi ikan ini nantinya akan digunakan sebagai hasil pemasukan daerah (PAD).

Permasalahan yang ada adalah jumlah hasil produksi tangkapan ikan ini tidak menentu untuk setiap bulannya yang mungkin dapat berdampak pada kegiatan produksi lain, seperti penyediaan es pendingin dan penyediaan air bersih

untuk proses produksi. Hal ini juga mungkin dapat berdampak akhir pada jumlah nilai pemasukan pendapatan daerah (PAD).

Jumlah produksi ikan ini akan diramalkan untuk satu periode kedepan dengan menggunakan data-data *history* yang sudah ada dengan menggunakan metode BPNN dan model optimal yang dihasilkan pada akhir penelitian dan hasil analisisnya dapat digunakan UPTD Pelabuhan Perikanan ini sendiri sebagai salah satu bahan acuan yang berhubungan dengan kegiatan ini.

1.2. Perumusan Masalah

Berdasarkan latar belakang permasalahan di atas, tugas akhir yang diajukan ini akan menitik-beratkan pada permasalahan seperti pada hal-hal berikut:

1. Bagaimana membangun model yang tepat untuk meramalkan data produksi ikan Pelabuhan Perikanan Kota Banjarmasin (PPKB) dengan BPNN?
2. Bagaimana hasil analisa BPNN terhadap produksi ikan PPKB pada periode-periode yang ditentukan?
3. Bagaimana perancangan aplikasi produksi ikan pada PPKB (Pelabuhan Perikanan Kota Banjarmasin)?

1.3. Batasan Masalah

Batasan-batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Data yang digunakan merupakan data jumlah produksi ikan dalam periode Januari 1998 hingga Agustus 2016 yang diambil dari laporan tahunan UPTD Pelabuhan Perikanan Kota Banjarmasin yang berjumlah 224 data sebagai *variabel x* dan data jumlah suhu udara pada stasiun Syamsudin Noor untuk jumlah dan periode yang sama yang diambil dari *website National Center for Environmental Information* sebagai *variabel y*.

2. Metode yang digunakan adalah *Backpropagation Neural Network*

1.4. Tujuan Tugas Akhir

Tujuan dari penelitian ini adalah untuk mendapatkan model, *output*, hasil analisa, dan aplikasi prediksi yang dapat digunakan oleh instansi bersangkutan.

1.5. Manfaat Tugas Akhir

Model, hasil analisa, dan aplikasi prediksi yang dihasilkan dalam penelitian ini mungkin dapat digunakan oleh instansi bersangkutan untuk memprediksi jumlah produksi ikan dalam rentang periode berikutnya di luar penelitian ini.

1.6. Relevansi Tugas Akhir

Penelitian ini berhubungan dengan bidang Teknik Peramalan, Penggalian Data dan Analisa Bisnis, dan Rancang Bangun Perangkat Lunak.

“Halaman ini sengaja dikosongkan”

BAB II

TINJAUAN PUSTAKA

Bab ini akan menjelaskan mengenai penelitian sebelumnya dan dasar teori yang dijadikan acuan dalam pengerjaan penelitian ini. Landasan teori akan memberikan gambaran secara umum dari landasan penelitian ini.

2.1. Studi Sebelumnya

Beberapa *paper* penelitian terkait yang dijadikan sebagai acuan dalam pengerjaan tugas akhir ini disajikan pada tabel 2.1, 2.2, 2.3, 2.4, 2.5, dan 2.6.

Pada *paper* 1 dijelaskan mengenai peramalan harga produksi listrik (SMP) menggunakan ANN 3 *layers* dengan kasus di Victorian Power System, Australia. Pada *paper* 2 dilakukan penelitian prediksi tingkah laku pelanggan dalam membeli item pada perusahaan ritel skala menengah dengan menggunakan metode gabungan MBA (*Market Basket Analysis*) dan ANN. Pada *paper* 3 dilakukan penelitian mengenai prediksi pola curah hujan bulanan dengan kasus Ketapang, Kalimantan Barat, dikatakan pada penelitian ini metode ANN belum tepat digunakan untuk memprediksi pola curah hujan. Pada *paper* 4 dilakukan penelitian mengenai pengaruh parameter pada peramalan produksi dan konsumsi hasil tambang, seperti gas alam dan minyak bumi. Pada *paper* 5 dilakukan penelitian untuk membandingkan dua metode (ARIMA dan ANN) dalam memproduksi barang-barang ritel di Eropa, dikatakan bahwa metode ARIMA lebih baik dibandingkan metode ANN dalam meramalkan produksi barang-barang tersebut. Dan pada *paper* 6 dilakukan perbandingan tiga metode dalam memproduksi hasil panen gandum di daerah-daerah tadah hujan di India. Dikatakan bahwa metode ARIMA menghasilkan peramalan yang buruk dalam memprediksi panen gandum di daerah-daerah tersebut dibandingkan dengan daerah dengan sistem irigasi.

Tabel 2. 1. Review Paper 1

Judul 1	<i>Electricity Price Short-Term Forecasting Using Artificial Neural Networks</i> [2]
Identitas	<ol style="list-style-type: none"> 1. B. R. Szkuta 2. L. A. Sanabria 3. T. S. Dillon 4. La Trobe University
Tujuan	Melakukan peramalan SMP (System Marginal Price) menggunakan 3 layers ANN Backpropagation
Hasil	<ul style="list-style-type: none"> <input type="checkbox"/> Berdasarkan uji sensitivitas, didapatkan bahwa variabel <i>input</i> berpengaruh terhadap variabel <i>output</i> yang dihasilkan <input type="checkbox"/> Berdasarkan perhitungan APE (<i>Absolute Percentage Error</i>) dan ESD (<i>Error Standard Deviation</i>) yang kecil, dikatakan bahwa hasil prediksi sangat memuaskan
Kesimpulan	<i>ANN Backpropagation</i> cukup baik digunakan untuk memprediksi harga produksi listrik dengan menggunakan parameter-parameter yang sesuai
Persamaan	Menggunakan metode yang sama, yaitu <i>Artificial Neural Network</i> dengan algoritma <i>Backpropagation</i>
Perbedaan	Perbedaan pada <i>data set</i> dan nilai-nilai parameter yang mungkin digunakan
Kelebihan	Penulis dapat menjelaskan dengan cukup baik penelitian ini kepada pembaca
Kekurangan	-

Tabel 2. 2. Review Paper 2

Judul 2	<i>Product Inventory Predictions at Small Medium Enterprise Using Market Basket Analysis Approach</i> [3]
Identitas	<ol style="list-style-type: none"> 1. Agus Mansur 2. Trioyoso Kuncoro 3. Universitas Islam Indonesia
Tujuan	Memahami perilaku pelanggan untuk memprediksi penjualan dengan menggunakan metode gabungan MBA dan ANN
Hasil	<ul style="list-style-type: none"> <input type="checkbox"/> Proses <i>training</i> hanya berjalan hingga 7 iterasi dan mendapatkan nilai <i>error</i> 0.000788252, sehingga dapat dikatakan performa model berjalan baik <input type="checkbox"/> Dari grafik yang dipaparkan, terdapat korelasi yang sangat baik antara data <i>target</i> dan data <i>output</i> pada jaringan <input type="checkbox"/> Didapatkan hasil prediksi 17 item dengan nilai prediksi tertinggi pada item “Black couplers K” dengan jumlah permintaan 1622 pada periode selanjutnya
Kesimpulan	<p>Berdasarkan hasil analisa didapatkan kesimpulan:</p> <ol style="list-style-type: none"> 1. Apabila pelanggan membeli item “klik klok” pasti juga akan membeli item “keyhole” dan “key 1.5 cm”. 2. Item “Black couplers K” merupakan item yang paling diminati pelanggan
Persamaan	Penggunaan salah satu metode yang sama, yaitu BPNN (<i>Backpropagation Neural Network</i>)
Perbedaan	<ul style="list-style-type: none"> • Pada paper menggunakan metode gabungan MBA (<i>Market Based Analysis</i>) dan ANN <i>Backpropagation</i>, sedangkan pada penelitian hanya menggunakan ANN <i>Backpropagation</i> • Perbedaan <i>dataset</i> yang digunakan
Kelebihan	Penulis dapat menjelaskan dengan baik kepada pembaca bagaimana metode gabungan tersebut dapat digunakan untuk melihat perilaku konsumen dalam membeli barang
Kekurangan	-

Tabel 2. 3. Review Paper 3

Judul 3	Metode Jaringan Saraf Tiruan Propagansi Balik Untuk Estimasi Curah Hujan Bulanan di Ketapang Kalimantan Barat [4]
Identitas	<ol style="list-style-type: none"> 1. Andi Ihwan 2. Universitas Tanjungpura
Tujuan	Memprediksi pola curah hujan bulanan di Ketapang, Kalimantan Barat
Hasil	<ul style="list-style-type: none"> <input type="checkbox"/> Tingkat korelasi data observasi dan data <i>training</i> saat proses pelatihan sangat baik. <input type="checkbox"/> Nilai sisa (<i>error</i>) dari data observasi dan data <i>testing</i> pada proses pengujian berbeda cukup jauh. <input type="checkbox"/> Perbandingan nilai data observasi dan data <i>forecast</i> memiliki perbedaan cukup jauh pada tahun 2012.
Kesimpulan	Metode ANN dengan arsitektur [15 10 5 1] dapat digunakan dalam pengenalan pola curah hujan bulanan di Kab. Ketapang. Apabila dilihat dari nilai residu yang dihasilkan (5,96 mm - 213,74 mm), metode ANN belum tepat untuk memprediksi pola curah hujan.
Persamaan	<ul style="list-style-type: none"> • Penggunaan metode yang sama, yaitu <i>Backpropagation Neural Network</i> • Tujuan yang ingin dicapai hampir mendekati, yaitu untuk melihat keakuratan metode BPNN dalam memprediksi suatu kasus
Perbedaan	Studi kasus yang diangkat berbeda dan pola data juga mungkin berbeda.
Kelebihan	Penulis memaparkan penjelasan penelitian dengan baik, sehingga memudahkan pembaca untuk memahami masalah yang dihadapi
Kekurangan	-

Tabel 2. 4. Review Paper 4

Judul 4	Analisis dan Implementasi Jaringan Syaraf Tiruan - Propagansi Balik Dalam Memprediksi Produksi dan Konsumsi Minyak Bumi, Gas Bumi, dan Batu Bara di Indonesia [5]
Identitas	<ol style="list-style-type: none"> 1. Anggit Nourislam 2. Jondri, Drs., MST 3. Siti Saadah, ST., MT. 4. Universitas Telkom
Tujuan	Mendapatkan model keluaran berupa nilai prediksi dari produksi dan konsumsi di masa mendatang dan dikelompokkan menurut indeks
Hasil	<ul style="list-style-type: none"> <input type="checkbox"/> Hasil terbaik pada <i>training</i> konsumsi batu bara adalah dengan <i>hidden layer</i> 6, penggunaan bias, dan <i>learning rate</i> 0,6 <input type="checkbox"/> Hasil terbaik pada <i>training</i> produksi batu bara adalah dengan <i>hidden layer</i> 6, penggunaan bias, dan <i>learning rate</i> 0,7 <input type="checkbox"/> Hasil terbaik pada <i>training</i> konsumsi gas bumi adalah dengan <i>hidden layer</i> 6, penggunaan bias, dan <i>learning rate</i> 0,4 <input type="checkbox"/> Hasil terbaik pada <i>training</i> produksi gas bumi adalah dengan <i>hidden layer</i> 8, penggunaan bias, dan <i>learning rate</i> 0,4 <input type="checkbox"/> Hasil terbaik pada <i>training</i> konsumsi minyak bumi adalah dengan <i>hidden layer</i> 6, penggunaan bias, dan <i>learning rate</i> 0,7 <input type="checkbox"/> Hasil terbaik pada <i>training</i> produksi minyak bumi adalah dengan <i>hidden layer</i> 6, penggunaan bias, dan <i>learning rate</i> 0,9

	<input type="checkbox"/> Hasil proses <i>testing</i> menghasilkan MAPE dibawah 10%, kecuali pada variabel produksi minyak bumi
Kesimpulan	<ul style="list-style-type: none"> • Semakin besar nilai <i>learning rate</i>, semakin cepat proses pembelajaran pada iterasi. Namun apabila <i>learning rate</i> terlalu besar, maka dapat mengakibatkan kondisi homogen data sulit tercapai • Penggunaan bias dapat memperkecil MAPE, karena membantu dalam mencapai kondisi homogen data • Semakin besar jumlah input belum tentu menghasilkan nilai MAPE yang semakin kecil • Dengan data yang selalu bergerak naik (tren naik), lebih baik menggunakan fungsi aktivasi linear karena pada fungsi aktivasi ini tidak membatasi nilai keluaran
Persamaan	<ul style="list-style-type: none"> • Penggunaan metode yang sama, yaitu Artificial Neural Network Backpropagation • Topik yang diangkat sama, yaitu mengenai peramalan produksi
Perbedaan	Tujuan penelitian berbeda dan pola data yang digunakan dimungkinkan berbeda
Kelebihan	Penulis memaparkan penjelasan penelitian dengan baik, sehingga memudahkan pembaca untuk memahami masalah yang dihadapi
Kekurangan	-

Tabel 2. 5. Review Paper 5

Judul 5	<i>Linear versus neural network forecast for European industrial production series</i> [6]
Identitas	<ol style="list-style-type: none"> 1. Saeed Heravi 2. Denise R. Osborn 3. C.R. Birchenhall
Tujuan	Membandingkan hasil prediksi menggunakan model linear (ARIMA) dan <i>neural network</i> (ANN) pada hasil produksi industri di Eropa
Hasil	Model <i>neural network</i> hanya memberikan nilai RMSE lebih rendah dibandingkan ARIMA hanya 5 dari 24 <i>series</i> (<i>variables</i>), yaitu pada periode 3 - 6 bulan ke depan
Kesimpulan	<ul style="list-style-type: none"> • Pemodelan ARIMA lebih baik digunakan untuk meramalkan jumlah produksi barang pada kasus ini dibandingkan metode ANN • Dikatakan bahwa pemodelan linear (ARIMA) akan berperan penting dalam kegiatan prediksi produksi barang-barang industri di Eropa
Persamaan	<ul style="list-style-type: none"> • Penggunaan salah satu metode yang sama, yaitu <i>Backpropagation Neural Network</i> • Topik yang diangkat sama, yaitu mengenai peramalan produksi
Perbedaan	Tujuan akhir penelitian berbeda
Kelebihan	Penulis memaparkan penjelasan penelitian dengan baik, sehingga memudahkan pembaca untuk memahami permasalahan yang dihadapi dan pemecahannya
Kekurangan	-

Tabel 2. 6. Review Paper 6

Judul 6	<i>Wheat production forecasting for a predominantly unirrigated region in Madhya Pradesh (India) [7]</i>
Identitas	<ol style="list-style-type: none"> 1. V. N. Sridhar 2. V. K. Dadhwal 3. K. N. Chaudhari 4. R. Sharma 5. G. D. Bairagi 6. A. K. Sharma
Tujuan	Membandingkan hasil prediksi produksi gandum menggunakan metode ARIMA dan dua metode lain dari penelitian sebelumnya
Hasil	<ul style="list-style-type: none"> □ Rata-rata hasil prediksi menggunakan metode <i>ARIMA</i>, <i>remote sensing</i>, <i>optimal estimate</i>, berturut-turut adalah 1153,4 kg ha⁻¹, 886,2 kg ha⁻¹, dan 962,6 kg ha⁻¹, sedangkan jumlah produksi gandum sebenarnya adalah 1020,5 kg ha⁻¹ □ Metode <i>optimal estimate</i> menghasilkan bias terendah (<i>Relative Deviation</i> = - 6.01%) dibandingkan dengan dua metode lainnya
Kesimpulan	<ul style="list-style-type: none"> • Metode ARIMA memberikan hasil peramalan yang buruk terhadap daerah-daerah tadah hujan, seperti Madhya Pradesh dan Rajashtan dibandingkan dengan daerah dengan sistem irigasi, seperti Punjab, Haryana, dan Uttar Pradesh barat. • Dikatakan pada penelitian bahwa metode ARIMA dapat menghasilkan prediksi yang lebih baik apabila menggunakan jumlah data lebih banyak, pada penelitian ini hanya menggunakan 35 periode (tahun) dan mempertimbangkan penggunaan variabel lain yang mempengaruhi pertumbuhan bibit gandum, seperti curah hujan
Persamaan	Topik sama, mengenai peramalan produksi

Perbedaan	Metode yang digunakan berbeda
Kelebihan	Penulis memaparkan penjelasan penelitian dengan baik, sehingga memudahkan pembaca untuk memahami permasalahan yang dihadapi dan pemecahannya
Kekurangan	-

2.2. Dasar Teori

Konsep atau teori yang dijadikan sebagai dasar acuan dalam penulisan tugas akhir ini dijelaskan pada bagian sub-bab Dasar Teori.

2.2.1. Dinas Perikanan dan Kelautan Provinsi Kalimantan Selatan

Dinas Perikanan dan Kelautan Provinsi Kalimantan Selatan merupakan salah satu badan pemasukan pendapatan negara yang dibawahhi oleh Kementerian Kelautan dan Perikanan Republik Indonesia. Diskanlut Prov. Kal-Sel ini terletak di Jalan Jend. Sudirman No. 9 Banjarbaru.

Dalam menyelesaikan tupoksi-tupoksi yang diberikan, terdapat beberapa UPT (Unit Pelaksana Teknis) yang dibentuk untuk membantu dalam penyelesaian tugas-tugas tersebut. Contohnya adalah UPTD Pelabuhan Perikanan Pantai Banjarmasin yang digunakan sebagai tempat pengambilan data dalam penelitian ini. Berikut adalah beberapa dari tupoksi Diskanlut Prov. Kal-Sel yang mungkin berhubungan dengan penelitian [7]:

- Perumusan kebijakan kelautan dan perikanan berdasarkan peraturan perundang-undangan;
- Pengkoordinasian penyelenggaraan urusan pemerintahan dan pelayanan umum di bidang kelautan dan perikanan;
- Penyelenggaraan urusan pemerintahan dan pelayanan umum di bidang kelautan dan perikanan;

- Pelaksanaan tugas di bidang kelautan, pesisir dan pulau-pulau kecil, perikanan tangkap, perikanan budidaya dan usaha kelautan dan perikanan;
- Pemantauan, evaluasi, dan pelaporan bidang kelautan dan perikanan.

2.2.2. UPTD Pelabuhan Perikanan Pantai Banjarmasin

Pelabuhan Perikanan Banjarmasin merupakan Unit Pelayanan Teknis Daerah (UPTD) Dinas Perikanan dan Kelautan Provinsi Kalimantan Selatan yang dibangun pada tahun 1975 dan merupakan salah satu Pelabuhan Perikanan tertua di Kalimantan. Dengan lokasi sangat strategis, sebagai pusat produksi dan merupakan tempat distribusi hasil perikanan tangkap ikan yang dapat diakses dari berbagai wilayah Kalimantan, Jawa, dan Sulawesi. Kalimantan Selatan memiliki luas perairan laut 127.536 hektare, dengan potensi produksi penangkapan perairan laut 180.000 ton dan memiliki banyak *fishing ground* potensial yang menjadi sasaran penangkapan. *Supply* ikan didapat dari kapal pengumpul dengan ukuran 6-30 GT yang mengambil ikan dari kapal penangkap di daerah Laut Jawa dan Selat Makassar. Sebagian diperoleh dari Truk dan Pickup yang membawa ikan dari Pelabuhan Batulicin dan Kotabaru.

UPTD Pelabuhan Perikanan Banjarmasin berlokasi 40 mil dari pemukiman nelayan di sungai barito, secara tepatnya di Kel. Pelambuan, Kota Banjarmasin, Kalimantan Selatan [8][9]. Salah satu tupoksi yang diberikan oleh Diskanlut Kal-Sel kepada UPT ini adalah memproduksi dan mendistribusikan hasil tangkapan ikan.

UPTD ini merupakan tempat penggalan data dari penelitian ini. Berikut adalah tupoksi yang dimiliki UPTD Pelabuhan Perikanan secara menyeluruh yang mungkin berhubungan dengan penelitian ini:

- a. Menyusun dan menetapkan program pengelolaan, pengembangan, pengendalian dan memelihara sarana pelabuhan perikanan;
- b. Mengelola, mengembangkan dan memelihara sarana pelabuhan perikanan;
- c. Memberikan layanan teknis kapal perikanan dan kesyahbandaran perikanan;
- d. Melaksanakan pengawasan administratif dan ketaatan kapal perikanan;
- e. Menyiapkan bahan dan melaksanakan penerapan sistem pusat pelabuhan perikanan (PIPP).

2.2.3. Variabel Pendukung (Suhu Udara)

Suhu udara digunakan sebagai variabel pendukung pada penelitian ini untuk melihat pengaruh suhu udara terhadap hasil produksi ikan dengan studi kasus PPPB.

2.2.3.1. Pengertian Suhu Udara

Suhu udara adalah ukuran panas dan dingin udara di atmosfer yang penyebarannya dibagi secara horisontal dan secara vertikal. Di laut, suhu bervariasi secara horisontal sesuai dengan garis lintang dan juga secara vertikal sesuai dengan kedalaman. Suhu merupakan salah satu faktor penting dalam mengatur proses kehidupan dan penyebaran organisme. Dikatakan suhu udara optimum untuk kehidupan kebanyakan organisme di muka bumi hanya berkisar antara 0-40°C. Selain itu, suhu juga sangat penting bagi kehidupan organisme di perairan, karena suhu mempengaruhi aktivitas dan perkembangbiakan organisme di perairan tersebut. Oleh karena itu, tidaklah heran jika banyak dijumpai bermacam-macam jenis ikan di berbagai tempat di dunia yang mempunyai toleransi tertentu terhadap perubahan suhu.

2.2.3.2. Pengaruh Suhu Udara terhadap Habitat Ikan

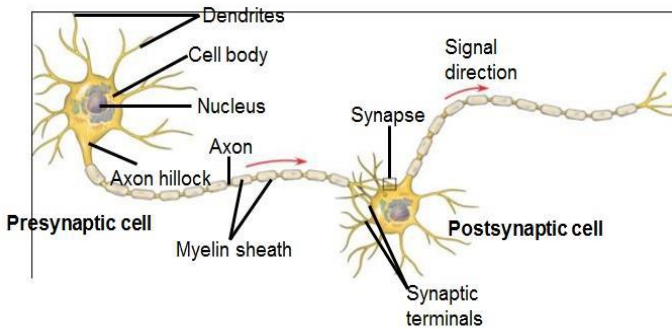
Menurut Hela dan Laevastu (1970), hampir semua populasi ikan yang hidup di laut mempunyai suhu optimum untuk kehidupannya [10], maka dengan mengetahui suhu optimum dari suatu spesies ikan, kita dapat menduga keberadaan kelompok ikan, yang kemudian dapat digunakan untuk tujuan perikanan.

Menurut Nybakken (1993), sebagian besar biota laut bersifat poikilometri (suhu tubuh dipengaruhi lingkungan), oleh karenanya pola penyebaran organisme di laut sangat mengikuti perbedaan suhu secara geografik [11].

Menurut Hela dan Laevastu (1970), pengaruh suhu terhadap ikan dapat dilihat dari proses metabolisme, seperti pertumbuhan, aktivitas tubuh, pengambilan makanan, kecepatan renang, dan juga saat proses pemijahan [10]. Suhu ekstrim pada suatu daerah tertentu selama musim pemijahan dapat memaksa ikan untuk bermigrasi mencari daerah dengan suhu optimum untuk dijadikan sebagai alternatif tempat memijah.

2.2.4. *Artificial Neural Network* (ANN)

Artificial Neural Network atau Jaringan Saraf Tiruan merupakan sebuah sistem pemrosesan data dengan meniru cara kerja sistem saraf manusia. Algoritma ANN lahir dari gagasan psikolog, Warren McCulloch dan Walter Pitts pada tahun 1943 yang menjelaskan cara kerja jaringan saraf dengan perangkat jaringan elektronik [12]. *Artificial Neural Network* adalah sebuah sistem yang terdiri atas banyak elemen pemrosesan sederhana yang terhubung secara paralel [13]. Cara kerja sistem saraf manusia dapat dilihat pada Gambar 2.1.



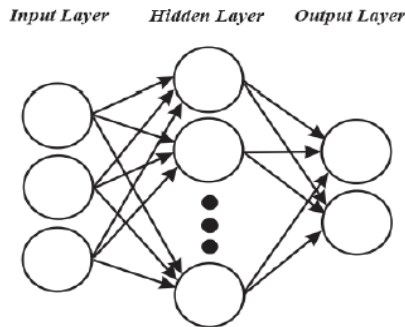
Gambar 2. 1. Jaringan Saraf Manusia

Seperti dijelaskan pada Gambar 2.1., *impulse* (rangsangan) pada jaringan saraf manusia diterima oleh *dendrit* dan kemudian diteruskan ke *axon* melalui badan sel (*cell body*). *Axon* akan mengantarkan rangsangan dari sel satu ke sel lainnya melalui *synapse* hingga diteruskan ke efektor. Setiap kegiatan diatur oleh *nucleus* di setiap sel.

ANN (*Artificial Neural Network*) terdiri dari sejumlah satuan masukan (*input*) dan keluaran (*output*) yang terkoneksi, dan pada setiap koneksinya terdapat bobot (*weight*) tersendiri yang dapat diubah-ubah untuk mendapatkan hasil prediksi yang diinginkan. Lapisan-lapisan pada ANN digambarkan pada Gambar 2.2.

- *Input Layer* (Lapisan Masukan): merupakan lapisan yang menghubungkan sumber data ke jaringan pemrosesan. Dalam artian, setiap masukan akan merepresentasikan variabel-variabel bebas yang berpengaruh terhadap keluaran (*output*)
- *Hidden Layer* (Lapisan Tersembunyi): merupakan lapisan perambatan variabel-variabel input untuk mendapatkan hasil *output* yang lebih mendekati keinginan. Suatu ANN *Multi Layer* dapat memiliki satu atau lebih *hidden layer*.

- *Output Layer* (Lapisan Keluaran): merupakan hasil keluaran dari pemrosesan data ANN. Keluaran yang didapatkan bergantung pada bobot, jumlah lapisan tersembunyi (*hidden layer*), dan fungsi aktivasi yang ditetapkan.



Gambar 2. 2. Arsitektur Multi Layer ANN [14]

Berdasarkan hubungan antara *layer/lapisan* yang digunakan untuk pemrosesan data, ANN dibagi menjadi tiga jenis jaringan, yaitu:

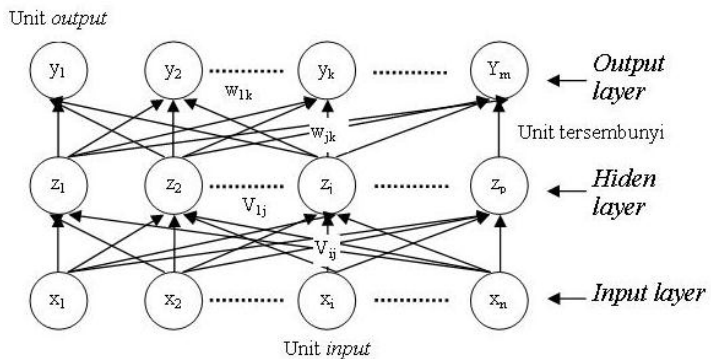
1. *Single Layer Network* adalah ANN yang tidak memiliki *hidden layer*, dalam artian *input layer* langsung terhubung ke *output layer*.
2. *Multi Layer Network* adalah ANN yang memiliki satu atau lebih *hidden layer* sebagai lapisan perambat.
3. *Competitive Layer Network* adalah ANN yang memiliki hubungan umpan balik (*feedback loop*) antara *output layer* dan *input layer*.

Dalam pengaplikasiannya ANN dibagi menjadi metode pembelajaran terawasi (*Supervised Learning*) dan metode pembelajaran tak terawasi (*Unsupervised Learning*), seperti penjelasan berikut:

1. *Supervised Learning* menggunakan sekumpulan data *input* dan terdapat contoh keluaran (*output/target*) yang diharapkan
2. *Unsupervised Learning* hanya menggunakan sekumpulan data *input* tanpa terdapat contoh keluaran (*output*)

2.2.4.1. Algoritma *Backpropagation*

Backpropagation merupakan salah satu algoritma yang digunakan pada *ANN Multi-Layer Network* dengan pembelajaran terawasi dimana dilakukan penyesuaian bobot secara berulang untuk mendapatkan nilai *error* terendah antara hasil prediksi dengan target yang diinginkan. *ANN Backpropagation* memiliki kelebihan yang bersifat *adaptive* dan *fault tolerance* terhadap pemecahan masalah dari sistem. Arsitektur Algoritma *Backpropagation* digambarkan pada Gambar 2.3.



Gambar 2. 3. Arsitektur ANN Backpropagation

Umumnya pada setiap metode peramalan, sebelum melakukan peramalan pada periode-periode kedepan, dilakukan dua proses pembelajaran terlebih dahulu, yaitu proses *training* dan *testing*. *Training* pada *ANN Backpropagation* merupakan proses pengaturan nilai

input serta bobot (*weight*) (seperti dijelaskan pada Gambar 2.3.) hingga didapatkan model yang optimal. *Training* pada *ANN Backpropagation* meliputi tiga fase, yaitu: fase pembelajaran (*learning step*), fase maju (*feed forward*), dan fase mundur (*backpropagation*) dari melihat *error-error* yang dihasilkan. *Testing* adalah pengujian kelayakan model yang telah diperoleh dari proses *training*.

Tahapan *training* pada *ANN Backpropagation* adalah sebagai berikut:

a. *Learning Step* (Tahap 1)

Tahapan pembelajaran (*learning step*) dilakukan dengan cara menginisiasi bobot awal kemudian mengulangi tahapan-tahapan yang ada secara berkala hingga kondisi akhir iterasi terpenuhi.

Untuk masing-masing *training data*, lakukan tahap 2 hingga tahap 7 hingga proses akhir iterasi terpenuhi.

b. *Fase Feedforward* (Tahap 2-4)

Tahapan ini dilakukan dengan mengalikan masing-masing *node* dengan *weight* (bobot) dan ditambahkan dengan biasnya hingga *output* didapatkan, tahapan ini bergerak maju dari *input layer* → *hidden layer* → *output layer*

Tahap 2

Masing-masing *node input* menerima sinyal masukan berupa X_i dan menyebarkan *node* tersebut ke bagian *hidden layer*

Tahap 3

Masing-masing *node* pada *hidden layer* dikalikan dengan bobot (*weight*) dan dijumlahkan dengan biasnya sesuai dengan persamaan (1)

$$Z_in_j = V_{0j} + \sum_{i=1}^n X_i V_{ij} \quad (1)$$

Dengan Z_in_j merupakan nilai keluaran untuk *node* Z_j , V_{0j} sebagai bobot (*weight*) pada bias *node* Z_j , X_i sebagai *node* ke- i pada *input layer*, dan V_{ij} sebagai bobot (*weight*) pada *node* X_i (*input layer*) dan *node* Z_j (*hidden layer*).

Setelah ditemukan nilai Z_in_j , langkah selanjutnya adalah menghitung nilai *node* pada lapisan tersembunyi (*hidden layer*) berdasarkan fungsi aktivasi yang digunakan sesuai dengan persamaan (2)

$$Z_j = f(Z_in_j) \quad (2)$$

Dengan Z_j adalah nilai pada *node* ke- j dan $f(Z_in_j)$ merupakan fungsi aktivasi dari Z_in_j . Keluaran dari fungsi aktivasi tersebut dikirim menuju *node* pada lapisan keluaran (*output layer*).

Tahap 4

Masing-masing *node* pada *output layer* dikalikan dengan bobot (*weight*) dan dijumlahkan dengan biasnya sesuai dengan persamaan (3)

$$Y_in_k = W_{0k} + \sum_{j=1}^p Z_j W_{jk} \quad (3)$$

Dengan Y_in_k merupakan nilai keluaran untuk *node* Y_k , W_{0k} sebagai bobot (*weight*) pada bias *node* Y_k , Z_j sebagai *node* ke- j pada *hidden layer*, dan W_{jk} sebagai bobot (*weight*) pada *node* Z_j (*hidden layer*) dan *node* Y_k (*output layer*).

Setelah ditemukan nilai Y_in_k , langkah selanjutnya adalah menghitung nilai *node* pada lapisan keluaran (*output layer*) berdasarkan fungsi aktivasi yang digunakan sesuai dengan persamaan (4)

$$Y_k = f(Y_in_k) \quad (4)$$

c. Fase Backpropagation (Tahap 5-6)

Pada tahapan ini dilakukan perhitungan nilai *error* dari keluaran pada *output layer* dibandingkan dengan pola masukan pada *input layer*, kemudian dilakukan perbaikan bobot pada setiap layer secara berkala

Tahap 5

Masing-masing *node* pada *output layer* menerima pola target sesuai dengan *input* pada tahapan *learning step*, kemudian dihitung nilai *error*-nya sesuai dengan persamaan (5)

$$\delta_k = (t_k - y_k)f'(y_in_k) \quad (5)$$

Dimana δ_k merupakan faktor pengendali nilai bobot (*weight*) pada lapisan luaran, $f'(y_in_k)$ merupakan turunan dari fungsi aktivasi pada y_in_k . Nilai t_k merupakan nilai target dari model, sehingga ditemukan selisih antara luaran dari tahapan *feedforward* dengan target yang diharapkan.

Kemudian menghitung perbaikan bobot (*weight*) dan memperbaiki nilai W_{jk} , sesuai dengan persamaan (6) dan (7)

$$\Delta W_{jk} = \alpha \delta_k Z_j \quad (6)$$

$$\Delta W_{0k} = \alpha \delta_k \quad (7)$$

Dimana ΔW_{jk} merupakan selisih antara W_{jk} saat t dengan W_{jk} saat $t+1$ dan α merupakan konstanta laju pelatihan (*learning rate*) dengan nilai $0 < \alpha < 1$.

Tahap 6

Masing-masing bobot (*weight*) yang menghubungkan *node* pada *output layer* dengan *node* pada *hidden layer* dikalikan delta (δ_k) dan dijumlahkan sebagai masukan (*input*) pada lapisan berikutnya, sesuai dengan persamaan (8)

$$\delta_in_j = \sum_{k=1}^m \delta_k W_{jk} \quad (8)$$

Dimana δ_in_j merupakan faktor pengendalian nilai bobot (*weight*) dari luaran lapisan tersembunyi (*hidden layer*). Selanjutnya, nilai dari δ_in_j dikalikan dengan turunan dari fungsi aktivasinya untuk menghitung nilai *error*-nya sesuai dengan persamaan (9)

$$\delta_j \delta_in_j f'(y_in_k) \quad (9)$$

Langkah berikutnya adalah menghitung perbaikan bobot (*weight*) yang digunakan untuk memperbaiki V_{ij} sesuai dengan persamaan (10) dan menghitung perbaikan bias untuk memperbaiki V_{oj} sesuai dengan persamaan (11)

$$\Delta V_{ij} = \alpha \delta_j X_i \quad (10)$$

$$\Delta V_{oj} = \alpha \delta_j \quad (11)$$

Tahap 7

Masing-masing luaran (*output*) dari *node* diperbaiki bias dan bobotnya sesuai dengan persamaan (12)

$$\begin{aligned} W_{jk}(\text{baru}) &= W_{jk}(\text{lama}) + \Delta W_{jk} \\ W_{ok}(\text{baru}) &= W_{ok}(\text{lama}) + \Delta W_{ok} \end{aligned} \quad (12)$$

Masing-masing *node* pada *hidden layer* diperbaiki bias dan bobotnya sesuai dengan persamaan (13)

$$V_{ij}(\text{baru}) = V_{ij}(\text{lama}) + \Delta V_{ij}$$

$$V_{oj}(\text{baru}) = V_{oj}(\text{lama}) + \Delta V_{oj} \quad (13)$$

Tahap 8

Uji kondisi akhir iterasi, berikut adalah kondisi dimana iterasi *backpropagation* dihentikan:

- Proses iterasi telah mencapai batas maksimal yang telah ditentukan
- *Error* yang didapatkan dapat ditoleransi
- Membuat *global minimum* dari SSE, menetapkan nilai *error* minimal

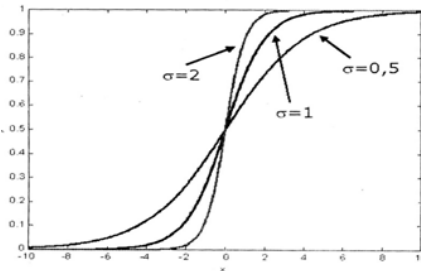
2.2.4.2. Fungsi Aktivasi BPNN

Fungsi Aktivasi BPNN merupakan suatu fungsi yang umumnya digunakan pada metode BPNN untuk mentransformasikan suatu *input* menjadi *output* tertentu. Fungsi Aktivasi BPNN antara lain:

Binary Sigmoid biasa digunakan untuk peramalan ANN dengan metode backpropagation. *Binary Sigmoid* mengubah masukan (*input*) yang merupakan variabel kontinu menjadi keluaran bernilai biner (0 dan 1). Formula perhitungan dengan aktivasi *binary sigmoid* digambarkan seperti pada gambar 2.4. dan persamaan (14)

$$y = f(x) = \frac{1}{1 + e^{-\sigma x}}$$

$$\text{dengan : } f'(x) = \sigma f(x) [1 - f(x)] \quad (14)$$

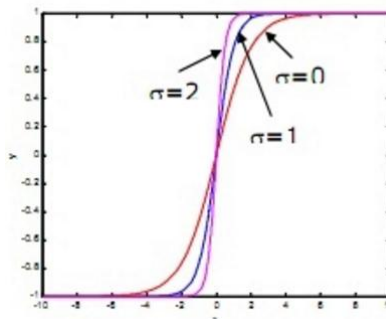


Gambar 2. 4. Fungsi *Binary Sigmoid*

Bipolar Sigmoid pada dasarnya hampir sama dengan fungsi sigmoid biner, akan tetapi fungsi ini mengubah variabel *input* menjadi variabel *output* dengan interval -1 hingga 1. Perhitungan fungsi *bipolar sigmoid* dapat dilihat pada persamaan (15) dan gambar 2.5.

$$y = f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$\text{dengan: } f'(x) = [1 + f(x)][1 - f(x)] \quad (15)$$

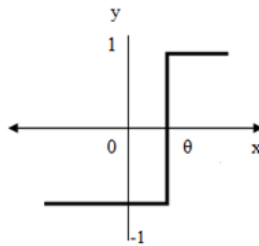


Gambar 2. 5. Fungsi *Bipolar Sigmoid*

***Bipolar* dengan nilai ambang (*threshold*)**, dimana nilai ambang menjadi garis pemisah antara daerah dengan respon aktivasi positif dan daerah dengan respon

aktivasi negatif, keluaran pada aktivasi ini dapat berupa nilai -1, 0, dan 1, seperti pada persamaan (16) dan gambar 2.6.

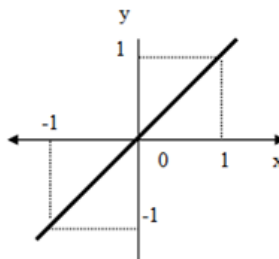
$$y = \begin{cases} 1 & \text{jika } x > 0 \\ 0 & \text{jika } x = 0 \\ -1 & \text{jika } x < 0 \end{cases} \quad (16)$$



Gambar 2. 6. Fungsi Bipolar Threshold

Linear Function digunakan apabila hasil keluaran (*output*) yang diinginkan sama dengan masukan (*input*) seperti pada persamaan (17) dan Gambar 2.7.

$$y = x \quad (17)$$



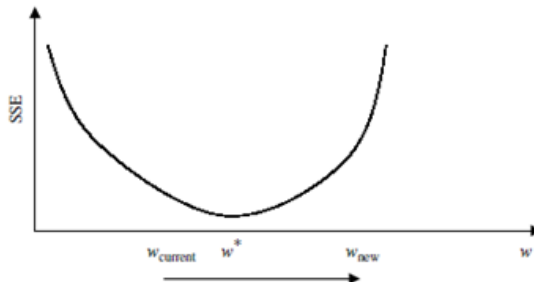
Gambar 2. 7 Fungsi Linear

2.2.4.2. Parameter BPNN

Parameter-parameter yang dapat digunakan dalam metode BPNN, antara lain:

Learning Rate

Learning Rate memiliki peran penting dalam hal waktu yang digunakan untuk mencapai minimum *error*. Dalam parameter ini digunakan skala dari 0 hingga 1, semakin besar nilai *learning rate*, maka waktu yang dibutuhkan pada proses *training* semakin singkat. Namun di sisi lain, apabila nilai *learning rate* terlalu besar, hasil proses *training* dapat melewati keadaan dimana nilai *error* minimal telah dicapai [15]. Pengaruh penggunaan parameter *learning rate* secara optimal digambarkan pada Gambar 2.8.



Gambar 2. 8. Gradient Descent

Berdasarkan Gambar 2.8., nilai optimal bobot yang memiliki error terendah didefinisikan sebagai w^* dan berdasarkan gambar tersebut dapat dikatakan bahwa nilai bobot baru (iterasi) yang didefinisikan sebagai w_{new} dapat melampaui nilai optimal. Oleh karena itu, diperlukan percobaan untuk menentukan nilai *learning rate* yang optimal.

Momentum

Momentum merupakan salah satu parameter yang digunakan dalam metode BPNN. *Momentum* berkisar antara skala 0 hingga 0.9. Penambahan *Momentum* dimaksudkan untuk menghindari perubahan bobot yang mencolok (*local minimum/local maximum*) akibat adanya *outlier* (data yang sangat berbeda).

Epoch (Lama Iterasi)

Jumlah maksimal iterasi yang digunakan mempengaruhi penentuan titik optimum bobot yang digunakan, akan tetapi semakin banyak *epoch* tidak semata-mata mempengaruhi semakin kecilnya *error* yang didapatkan pada proses perambatan ulang.

2.2.5. Performa Peramalan

Pengukuran performa peramalan penggunaan ANN dapat diukur dengan melihat skor *Mean Absolute Percentage Error* (MAPE) hasil peramalan. MAPE dihitung dengan menjumlahkan setiap kesalahan absolut pada tiap periode dengan mengurangkan nilai aktual dengan nilai hasil peramalan dibagi dengan nilai aktual periode tersebut baru kemudian dipresentasikan.

Skala pengkategorian nilai MAPE yang digunakan pada penelitian dan formula perhitungannya dijelaskan pada Tabel 2.7., Tabel 2.8. dan persamaan (18):

$$MAPE = \left(\frac{100}{n}\right) \sum \left| \frac{At - Ft}{At} \right| \quad (18)$$

Keterangan:

- At : permintaan aktual pada periode ke-t
- Ft : peramalan permintaan pada periode ke-t
- n : jumlah periode peramalan yang terlibat

Tabel 2. 7. Skala Pengukuran Kinerja Model

MAPE	Hasil Peramalan
<10%	Sangat Baik
10-20%	Baik
20-50%	Layak/ Cukup
>50%	Buruk

Seperti dijelaskan pada Tabel 2.7., suatu model dikatakan memiliki kinerja ‘sangat baik’ apabila memiliki nilai MAPE dibawah 10% dan memiliki kinerja ‘baik’ apabila nilai MAPE berkisar antara 10% - 20% dan dikatakan ‘layak’ apabila nilai MAPE berkisar antara 20% - 50% dan apabila lebih dari itu dikatakan berkinerja buruk [16].

Tabel 2. 8. Skala Pengukuran Kinerja Variabel

Skala MAPE	Performa Variabel Peramalan
0.1% - 5%	Kurang mempengaruhi
5.1% – 10%	Cukup mempengaruhi
10.1 – 20%	Mempengaruhi
20.1% – 50%	Sangat mempengaruhi
> 50%	Amat sangat mempengaruhi

Seperti dijelaskan pada Tabel 2.8., suatu variabel atau parameter dikatakan kurang mempengaruhi hasil akhir peramalan apabila menghasilkan perbedaan MAPE hanya berkisar antara 0.1% - 5%, dikatakan cukup mempengaruhi apabila menghasilkan perbedaan MAPE berkisar antara 5.1% - 10%, dikatakan mempengaruhi apabila MAPE berbeda sebanyak 10.1% - 20%, dikatakan sangat mempengaruhi apabila menghasilkan perbedaan MAPE 20.1% - 50%, dan dikatakan amat sangat mempengaruhi apabila menghasilkan perbedaan MAPE lebih dari 50%.

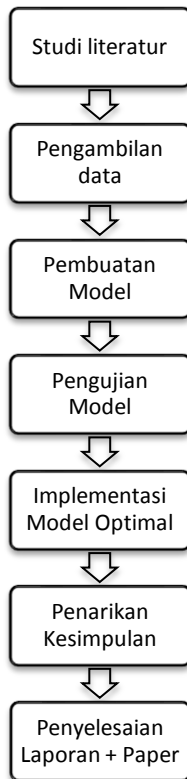
BAB III

METODOLOGI PENELITIAN

Bab Metodologi Penelitian merupakan penjelasan alur tahapan pengerjaan tugas akhir. Tahapan pada metodologi penelitian digunakan sebagai panduan pengerjaan tugas akhir agar dapat diselesaikan secara terarah, dan teratur, serta sistematis.

3.1. Tahapan Pelaksanaan Tugas Akhir

Tahapan pelaksanaan dari penelitian ini digambarkan pada Gambar 3.1. Tahap-tahap pelaksanaan ini akan dijelaskan pada sub-bab selanjutnya.



Gambar 3. 1. Metodologi Penelitian

3.1.1. Studi Literatur

Tahap ini merupakan tahapan pembelajaran terkait dengan topik yang diangkat sebagai penelitian. Tahapan ini terkait dengan proses-proses seperti pembelajaran materi ANN, tahapan-tahapan pada metode ANN, dan melakukan *review* paper-paper terkait penggunaan metode ANN *Backpropagation* (BPNN).

3.1.2. Pengambilan Data

Tahapan ini adalah proses pengambilan data yang digunakan untuk penelitian. Data yang diambil merupakan data produksi ikan UPTD Pelabuhan Perikanan Banjarmasin, Dinas Perikanan dan Kelautan Provinsi Kalimantan Selatan periode Januari 1998 hingga Agustus 2016.

3.1.3. Pembuatan Model

Tahapan penentuan parameter dan pembuatan model menggunakan parameter untuk nantinya digunakan pada proses *training* dan proses *testing*.

3.1.4. Pengujian Model

Tahap pengujian model setelah melalui proses *training* . Hal ini bertujuan untuk melihat performa model dalam menggunakan *dataset* yang berbeda..

3.1.5. Implementasi Model Optimal

Setelah didapatkan model optimal melalui proses *training* dan *testing*. Maka dengan model optimal ini dilakukan peramalan untuk satu periode kedepan dan analisa lanjut erupakan tahapan implementasi dan tahap perancangan aplikasi setelah model yang optimal didapat.

3.1.6. Penarikan Kesimpulan

Tahapan selanjutnya adalah penarikan kesimpulan akhir yang didapatkan berdasar dari proses implementasi, seperti apakah metode BPNN baik digunakan pada data berpola seperti ini, apakah penentuan parameter sangat berpengaruh pada hasil akhir peramalan, dan sebagainya.

3.1.7. Penyelesaian Laporan dan Paper

Tahapan terakhir adalah penyelesaian laporan dan paper terkait dengan penelitian berdasarkan implementasi dan hasil analisa yang didapat.

- **Bab I Pendahuluan**
Pada bab ini dijelaskan mengenai uraian hasil identifikasi masalah yang ingin dipecahkan yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat hasil penelitian, serta relevansi penelitian.
- **Bab II Dasar Teori**
Bab ini menjelaskan mengenai penelitian-penelitian terkait dan landasan-landasan teori yang dijadikan acuan pada penelitian peramalan produksi ikan ini.
- **Bab III Metodologi Penelitian**
Bab ini menjelaskan mengenai alur pengerjaan dari penelitian, yaitu dari proses studi literatur hingga penyelesaian laporan tugas akhir.
- **Bab IV Perencanaan**
Bab ini menjelaskan tahapan perancangan sebelum implementasi dilakukan. Obyek apa saja yang digunakan akan dijelaskan pada tahapan perancangan.

- Bab V Implementasi

Bab ini berisi mengenai penjelasan proses implementasi hasil dari tahapan perancangan, seperti penggunaan tool Matlab dan penjelasan *syntax* perancangan aplikasi.

- Bab VI Hasil dan Pembahasan

Bab ini berisi mengenai penjelasan hasil proses implementasi dan kemudian akan dilakukan analisa berdasarkan hasil yang diperoleh.

- Bab VII Kesimpulan dan Saran

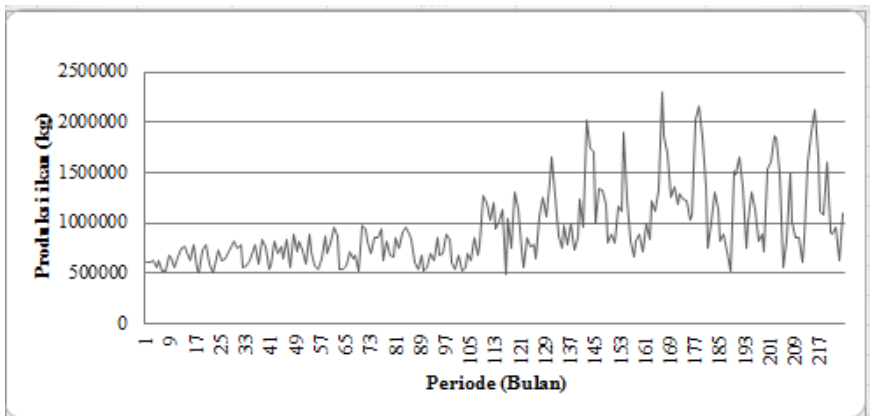
Bab ini berisi kesimpulan akhir hasil penelitian dan saran yang dapat disampaikan untuk penelitian selanjutnya.

BAB IV PERANCANGAN

Pada bab ini akan dijelaskan mengenai apa saja obyek perancangan awal yang nantinya akan digunakan pada pengimplementasian Bab V.

4.1. Data Input

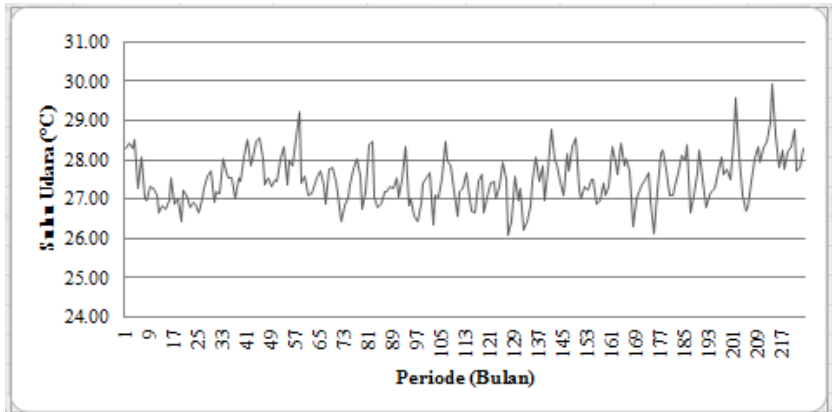
Data yang digunakan diambil dari data laporan tahunan Dinas Perikanan dan Kelautan Prov. Kalsel, yaitu data jumlah produksi ikan instansi tersebut (sebagai variabel x) dan data yang diunduh secara online pada *website* National Center For Environmental Information (NCEI) [17], sebagai tempat pengambilan data jumlah suhu udara pada stasiun Syamsudin Noor (sebagai variabel y), seperti dijelaskan pada Gambar 4.1. & Gambar 4.2. Keseluruhan data dibagi menjadi rasio 70% untuk data *training* dan 30% untuk data *testing*, yaitu 156 periode sebagai data *training* dan 68 periode sebagai data *testing*.



Gambar 4. 1. Pola Data Variabel Produksi Ikan

Pada Gambar 4.1. dapat dilihat bahwa *dataset* jumlah produksi ikan berpola horisontal dari periode 1 hingga 121, sedangkan dari periode 122 hingga 224 dapat dilihat sekilas

bahwa *dataset* berubah pola menjadi berpola musiman (*seasonal*).



Gambar 4. 2. Pola Data Variabel Suhu Udara

Pada Gambar 4.2. dapat dilihat bahwa data rata-rata suhu udara pada stasiun Syamsudin Noor berpola horisontal (fluktuatif) dari awal periode hingga akhir periode, yaitu periode 1 hingga 224.

4.2. Model ANN (*Artificial Neural Network*)

Setelah melalui proses diskusi dengan pembimbing, maka diambil keputusan bahwa keseluruhan model akan dibagi menjadi 10 dari *range* ($n - 1$ periode) hingga ($n - 5$ periode) dengan menggunakan 2 variabel yang ada, yaitu jumlah produksi ikan dan rata-rata suhu udara pada stasiun Syamsudin Noor. Seluruh model akan menggunakan jumlah *neuron hidden layer* dengan formula m , $2m$, dan $3m$.

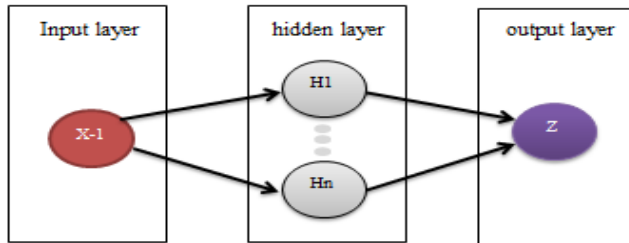
Keterangan:

n = periode sekarang

m = jumlah *neuron* masukan/*input*

4.2.1. Model 1

Model ini menggunakan 1 variabel masukan dengan *neuron* masukan ($x-1$), yaitu jumlah produksi ikan PPKB satu periode sebelumnya, seperti digambarkan pada Gambar 4.3.



Gambar 4. 3. Model ANN 1

Keterangan

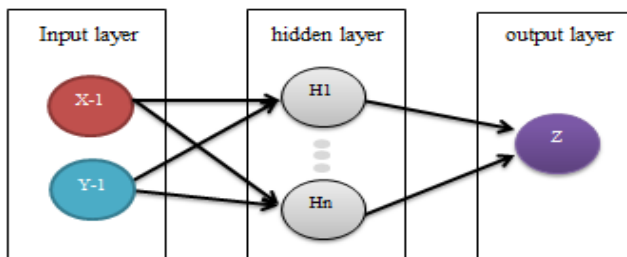
x = *neuron* masukan (jumlah produksi ikan)

h = *neuron hidden layer*

z = *neuron output*

4.2.2. Model 2

Model ini menggunakan 2 variabel masukan dengan *neuron* masukan ($x-1$) dan ($y-1$), yaitu jumlah produksi ikan satu periode sebelumnya dan rata-rata suhu udara stasiun Syamsudin Noor satu periode sebelumnya, seperti digambarkan pada Gambar 4.4.



Gambar 4. 4. Model ANN 2

Keterangan:

x = *neuron* masukan (jumlah produksi ikan)

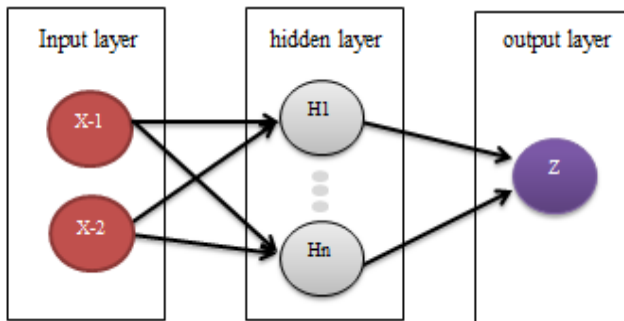
y = *neuron* masukan (rata-rata suhu udara)

h = *neuron hidden layer*

z = *neuron output*

4.2.3. Model 3

Model ini menggunakan 1 variabel masukan dengan masukan ($x-1$) dan ($x-2$), yaitu jumlah produksi ikan PPKB satu periode sebelumnya dan dua periode sebelumnya seperti digambarkan pada Gambar 4.5.



Gambar 4. 5. Model ANN 3

Keterangan:

x = *neuron* masukan (jumlah produksi ikan)

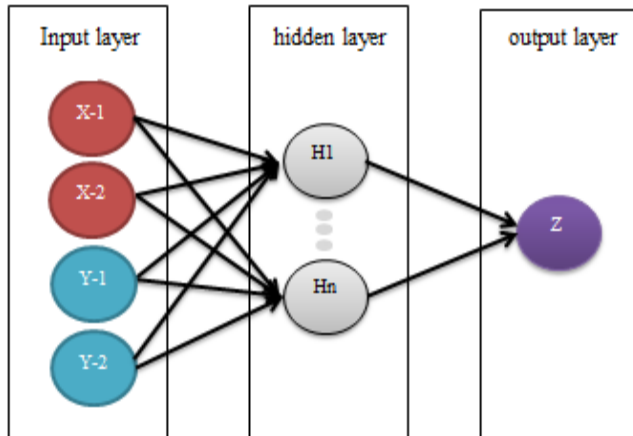
h = *neuron hidden layer*

z = *neuron output*

4.2.4. Model 4

Model ini menggunakan 2 variabel masukan dengan *neuron* masukan ($x-1$), ($x-2$), ($y-1$) dan ($y-2$), yaitu jumlah produksi ikan PPKB satu periode sebelumnya dan dua periode sebelumnya, dan jumlah rata-rata suhu udara stasiun

Syamsudin Noor satu periode sebelumnya dan dua periode sebelumnya seperti digambarkan pada Gambar 4.6.



Gambar 4. 6. Model ANN 4

Keterangan:

x = *neuron* masukan (jumlah produksi ikan)

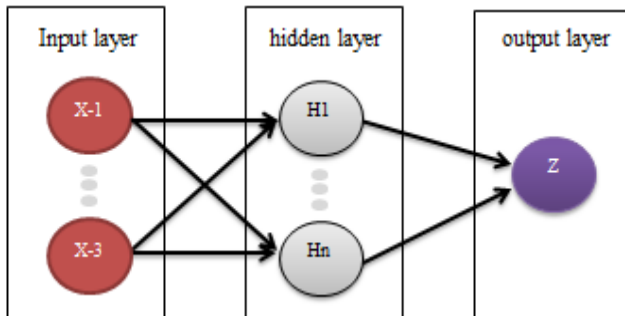
y = *neuron* masukan (rata-rata suhu udara)

h = *neuron hidden layer*

z = *neuron output*

4.2.5. Model 5

Model ini menggunakan 1 variabel masukan dengan *neuron* masukan ($x-1$), ($x-2$), dan ($x-3$), yaitu jumlah produksi ikan PPKB pada satu periode sebelumnya, dua periode sebelumnya, dan tiga periode sebelumnya seperti digambarkan pada Gambar 4.7.



Gambar 4. 7. Model ANN 5

Keterangan

x = *neuron* masukan (jumlah produksi ikan)

h = *neuron hidden layer*

z = *neuron output*

4.2.6. Model 6

Model ini menggunakan 2 variabel masukan dengan *neuron* masukan $(x-1)$, $(x-2)$, $(x-3)$, $(y-1)$, $(y-2)$ dan $(y-3)$, yaitu jumlah produksi ikan PPKB pada satu periode sebelumnya, dua periode sebelumnya, dan tiga periode sebelumnya, dan rata-rata suhu udara stasiun Syamsudin Noor pada satu periode sebelumnya, dua periode sebelumnya, dan tiga periode sebelumnya, seperti digambarkan pada Gambar 4.8.

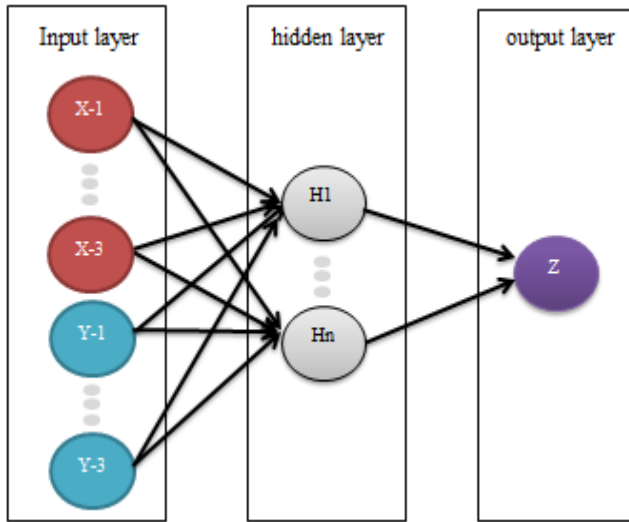
Keterangan:

x = *neuron* masukan (jumlah produksi ikan)

y = *neuron* masukan (rata-rata suhu udara)

h = *neuron hidden layer*

z = *neuron output*



Gambar 4. 8. Model ANN 6

4.2.7. Model 7

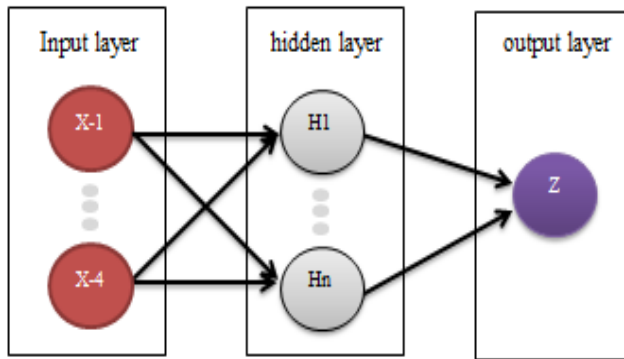
Model ini menggunakan 1 variabel masukan dengan *neuron* masukan ($x-1$), ($x-2$), ($x-3$) dan ($x-4$), yaitu jumlah produksi ikan PPKB pada satu periode sebelumnya, dua periode sebelumnya, tiga periode sebelumnya, dan empat periode sebelumnya, seperti digambarkan pada Gambar 4.9.

Keterangan

x = *neuron* masukan (jumlah produksi ikan)

h = *neuron hidden layer*

z = *neuron output*



Gambar 4. 9. Model ANN 7

4.2.8. Model 8

Model ini menggunakan 2 variabel masukan dengan *neuron* masukan (x-1), (x-2), (x-3), (x-4), (y-1), (y-2), (y-3) dan (y-4), yaitu jumlah produksi ikan PPKB pada satu periode sebelumnya, dua periode sebelumnya, tiga periode sebelumnya, dan empat periode sebelumnya, dan rata-rata suhu udara stasiun Syamsudin Noor pada satu periode sebelumnya, dua periode sebelumnya, tiga periode sebelumnya, dan empat periode sebelumnya, seperti digambarkan pada Gambar 4.10.

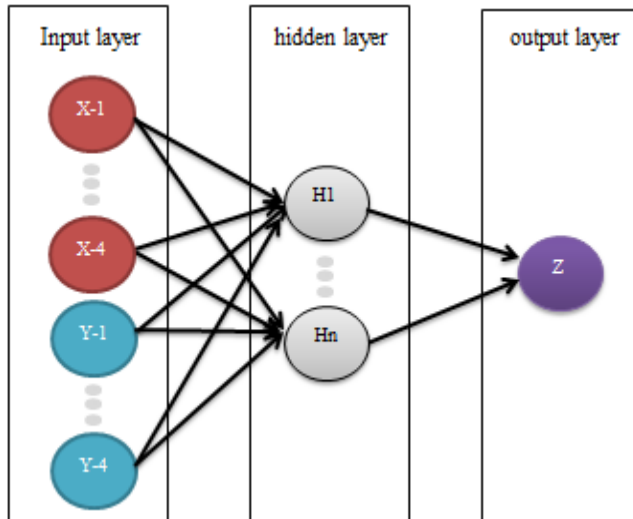
Keterangan:

x = *neuron* masukan (jumlah produksi ikan)

y = *neuron* masukan (rata-rata suhu udara)

h = *neuron hidden layer*

z = *neuron output*



Gambar 4. 10. Model ANN 8

4.2.9 Model 9

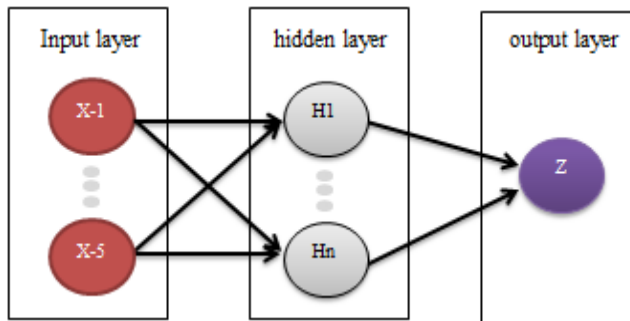
Model ini menggunakan 1 variabel masukan dengan *neuron* masukan ($x-1$), ($x-2$), ($x-3$), ($x-4$), dan ($x-5$), yaitu jumlah produksi ikan PPKB satu periode sebelumnya, dua periode sebelumnya, tiga periode sebelumnya, empat periode sebelumnya, dan lima periode sebelumnya, seperti digambarkan pada Gambar 4.11.

Keterangan

x = *neuron* masukan (jumlah produksi ikan)

h = *neuron hidden layer*

z = *neuron output*



Gambar 4. 11. Model ANN 9

4.2.10. Model 10

Model ini menggunakan 2 variabel masukan dengan *neuron* masukan ($x-1$), ($x-2$), ($x-3$), ($x-4$), ($x-5$), ($y-1$), ($y-2$), ($y-3$), ($y-4$), dan ($y-5$), yaitu jumlah produksi ikan PPKB satu periode sebelumnya, dua periode sebelumnya, tiga periode sebelumnya, empat periode sebelumnya, dan lima periode sebelumnya, serta rata-rata suhu udara stasiun Syamsudin Noor pada satu periode sebelumnya, dua periode sebelumnya, tiga periode sebelumnya, empat periode sebelumnya, dan lima periode sebelumnya, seperti digambarkan pada Gambar 4.12.

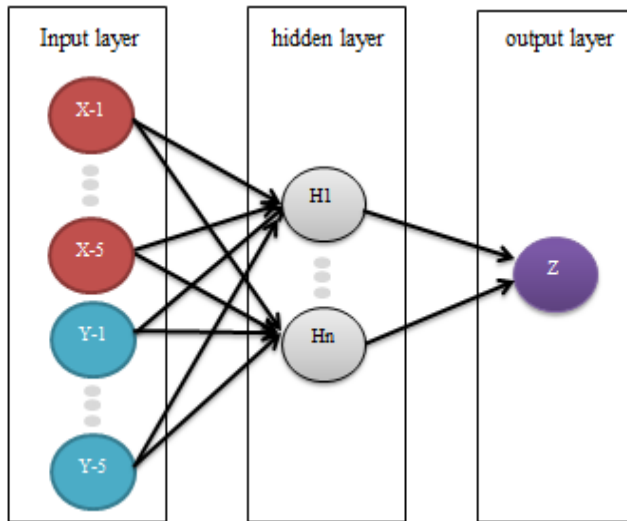
Keterangan:

x = *neuron* masukan (jumlah produksi ikan)

y = *neuron* masukan (rata-rata suhu udara)

h = *neuron hidden layer*

z = *neuron output*



Gambar 4. 12. Model ANN 10

4.3. Perancangan Aplikasi

Pada sub-bab ini dijelaskan mengenai perancangan awal untuk pembuatan aplikasi peramalan sederhana pada studi kasus UPTD Pelabuhan Perikanan Banjarmasin.

4.3.1. Kebutuhan Fungsional

Pada sub-bab ini dijelaskan mengenai kebutuhan fungsional pada perancangan aplikasi sederhana UPTD Pelabuhan Perikanan Banjarmasin.

4.3.1.1. Fungsi Menampilkan Data History

Fungsi ini digunakan untuk melihat periode, data jumlah produksi ikan PPKB, dan suhu udara stasiun Syamsudin Noor pada keseluruhan periode, yaitu berjumlah 224 periode.

4.3.1.2. Fungsi Training

Fungsi ini memungkinkan pengguna aplikasi untuk melakukan proses *training* dengan rasio 70% pada keseluruhan *data set* yang digunakan, yaitu berjumlah 156 periode.

4.3.1.3. Fungsi Testing

Fungsi ini memungkinkan pengguna aplikasi untuk melakukan proses *testing* dengan rasio 30% pada keseluruhan *data set* yang digunakan, yaitu berjumlah 68 periode.

4.3.1.4. Fungsi Menampilkan Testing Error

Fungsi ini memungkinkan pengguna aplikasi untuk melihat hasil *testing error* dari proses *testing* dengan menggunakan model optimal yang didapatkan dari hasil proses *training*.

4.3.1.5. Fungsi Forecast

Fungsi ini memungkinkan pengguna aplikasi untuk meramalkan jumlah produksi ikan PPKB (Pelabuhan Perikanan Banjarmasin) untuk beberapa periode ke depan setelah pengguna aplikasi selesai melakukan proses *training* dan *testing*. Dan hasilnya nantinya dapat dilihat pengguna aplikasi dalam bentuk tabel.

4.3.1.6. Fungsi Menampilkan Grafik

Fungsi ini memungkinkan pengguna aplikasi untuk melihat keseluruhan *data history* yang berjumlah 224 dan hasil peramalan sesuai dengan periode yang diramalkan melalui grafik.

4.3.2. Use Case Description

Pada sub-bab ini akan dijelaskan mengenai prosedur penggunaan aplikasi oleh pengguna aplikasi yang fungsional-fungsional telah dijelaskan pada sub-bab sebelumnya.

4.3.2.1. Use Case Menampilkan Data History

Alur untuk melihat keseluruhan *data history* UPTD Pelabuhan Perikanan Banjarmasin oleh pengguna aplikasi dapat dilihat pada Tabel 4.1.

Tabel 4. 1. Use Case Menampilkan Data History

<i>Use Case Name</i>	Menampilkan Data History
<i>Purpose</i>	Untuk melihat data <i>history</i> jumlah produksi ikan PPKB dan rata-rata suhu udara pada stasiun Syamsudin Noor
<i>Actors</i>	Pengguna aplikasi
<i>Pre-Condition</i>	<ul style="list-style-type: none"> Pada komputer pengguna harus terpasang <i>Java Runtime Environment</i> (JRE) minimal versi 8.111
<i>Post-Condition</i>	Pengguna dapat melihat data <i>history</i> produksi ikan PPKB dan rata-rata suhu udara pada stasiun Syamsudin Noor
<i>Flow events</i>	<ol style="list-style-type: none"> Pengguna membuka Aplikasi Peramalan Produksi Ikan UPTD Pelabuhan Perikanan Banjarmasin Sistem menampilkan jendela mengenai data <i>history</i> produksi ikan PPKB dan rata-rata suhu udara stasiun Syamsudin Noor
<i>Alternate Flow Events</i>	<ol style="list-style-type: none"> Pengguna telah berada dalam aplikasi peramalan, untuk membuka kembali tampilan data <i>history</i> dapat dengan menekan <i>tab</i> 'Data Aktual'
<i>Exceptional Flow events</i>	<ul style="list-style-type: none"> Komputer pengguna mati daya karena alasan tertentu, aplikasi akan tertutup secara otomatis

4.3.2.2. Use Case Training Process

Alur untuk melakukan proses *training* data oleh pengguna aplikasi dengan menggunakan *dataset* yang telah ditentukan yang berjumlah 156 periode dapat dilihat pada Tabel 4.2.

Tabel 4. 2. Use Case Training Process

<i>Use Case Name</i>	Training Process
<i>Purpose</i>	Untuk melakukan proses <i>training</i> menggunakan model optimum pada <i>train dataset</i>
<i>Actors</i>	Pengguna aplikasi
<i>Pre-Condition</i>	<ul style="list-style-type: none"> • Pada komputer pengguna harus terpasang <i>Java Runtime Environment</i> (JRE) minimal versi 8.111
<i>Post-Condition</i>	Pengguna telah melakukan proses <i>training</i> menggunakan <i>train dataset</i>
<i>Flow events</i>	<ol style="list-style-type: none"> 1. Pengguna membuka Aplikasi Peramalan Produksi Ikan UPTD Pelabuhan Perikanan Banjarmasin 2. Sistem menampilkan jendela mengenai data <i>history</i> produksi ikan PPKB dan rata-rata suhu udara stasiun Syamsudin Noor 3. Pengguna menekan tombol ‘Train Model’ pada bagian bawah kanan jendela 4. Sistem melakukan proses <i>training</i> menggunakan data <i>train set</i>
<i>Alternate Flow Events</i>	<ol style="list-style-type: none"> 1. Pengguna telah berada dalam aplikasi peramalan, untuk melakukan proses <i>training</i> dapat dengan menekan tombol ‘Train Model’ pada <i>tab</i> ‘Data Aktual’
<i>Exception al Flow events</i>	<ul style="list-style-type: none"> • Komputer pengguna mati daya karena alasan tertentu, aplikasi akan tertutup secara otomatis

4.3.2.3. Use Case Testing Process

Alur untuk melakukan proses *testing* data oleh pengguna aplikasi dengan menggunakan *dataset* yang telah ditentukan yang berjumlah 68 periode dapat dilihat pada Tabel 4.3.

Tabel 4. 3. Use Case Testing Process

<i>Use Case Name</i>	Testing Process
<i>Purpose</i>	Untuk melakukan proses <i>testing</i> menggunakan model optimum pada <i>test dataset</i>
<i>Actors</i>	Pengguna aplikasi
<i>Pre-Condition</i>	<ul style="list-style-type: none"> Pada komputer pengguna harus terpasang <i>Java Runtime Environment</i> (JRE) minimal versi 8.111
<i>Post-Condition</i>	Pengguna telah melakukan proses <i>training</i> menggunakan <i>train dataset</i>
<i>Flow events</i>	<ol style="list-style-type: none"> Pengguna membuka Aplikasi Peramalan Produksi Ikan UPTD Pelabuhan Perikanan Banjarmasin Sistem menampilkan jendela mengenai data <i>history</i> produksi ikan PPKB dan rata-rata suhu udara stasiun Syamsudin Noor Pengguna menekan tombol ‘Train Model’ pada bagian bawah kanan jendela Sistem melakukan proses <i>training</i> menggunakan data <i>train set</i> dan menampilkan notifikasi Pengguna menekan tombol ‘OK’ dan kemudian menekan tombol ‘Test Model’ pada bagian bawah kanan jendela Sistem melakukan proses <i>testing</i> menggunakan data <i>test set</i>
<i>Alternate Flow Events</i>	<ul style="list-style-type: none"> Pengguna telah berada dalam aplikasi peramalan, untuk melakukan proses <i>testing</i> dapat dengan menekan tombol ‘Test Model’ pada <i>tab</i> ‘Data Aktual’
<i>Exceptional Flow events</i>	<ul style="list-style-type: none"> Komputer pengguna mati daya karena alasan tertentu, aplikasi akan tertutup secara otomatis

4.3.2.4. Use Case Testing Error

Alur kegiatan untuk melihat hasil *testing error* setelah melakukan proses *training* dan *testing* oleh pengguna dapat dilihat pada Tabel 4.4.

Tabel 4. 4. Use Case Testing Error

<i>Use Case Name</i>	Menampilkan Testing Error
<i>Purpose</i>	Untuk melihat hasil <i>error</i> proses <i>testing</i> menggunakan model optimum
<i>Actors</i>	Pengguna aplikasi
<i>Pre-Condition</i>	<ul style="list-style-type: none"> • Pada komputer pengguna harus terpasang <i>Java Runtime Environment</i> (JRE) minimal versi 8.111 • Pengguna telah melakukan proses <i>training</i> dan proses <i>testing</i>
<i>Post-Condition</i>	Pengguna dapat melihat hasil <i>error</i> proses <i>testing</i>
<i>Flow events</i>	<ol style="list-style-type: none"> 1. Pengguna menekan <i>tab</i> ‘Tabel Hasil Peramalan’ pada aplikasi 2. Sistem menampilkan jendela ‘Tabel Hasil Peramalan’ 3. Pengguna melihat hasil <i>test error</i> pada bagian kanan atas jendela yaitu dengan tulisan ‘Model Error (MAPE)’
<i>Alternate Flow Events</i>	<ul style="list-style-type: none"> • Pengguna telah berada dalam aplikasi peramalan, untuk melihat hasil <i>test error</i> dapat dengan menekan <i>tab</i> ‘Tabel Hasil Peramalan’
<i>Exception al Flow events</i>	<ul style="list-style-type: none"> • Komputer pengguna mati daya karena alasan tertentu, aplikasi akan tertutup secara otomatis

4.3.2.5. Use Case Forecast

Alur penggunaan untuk melakukan fungsi forecast oleh pengguna setelah melalui proses *testing* dapat dilihat pada Tabel 4.5.

Tabel 4. 5. Use Case Forecast

<i>Use Case Name</i>	Forecast
<i>Purpose</i>	Untuk melakukan <i>forecasting</i> menggunakan model optimum
<i>Actors</i>	Pengguna aplikasi
<i>Pre-Condition</i>	<ul style="list-style-type: none"> • Pada komputer pengguna harus terpasang <i>Java Runtime Environment</i> (JRE) minimal versi 8.111 • Pengguna telah melakukan proses <i>training</i> dan proses <i>testing</i>
<i>Post-Condition</i>	Pengguna telah melakukan proses peramalan menggunakan model optimal
<i>Flow events</i>	<ol style="list-style-type: none"> 1. Pengguna menekan <i>tab</i> ‘Tabel Hasil Peramalan’ pada aplikasi 2. Sistem menampilkan jendela ‘Tabel Hasil Peramalan’ 3. Pengguna memasukkan jumlah periode yang ingin diramalkan (hanya <i>integer</i>) 4. Pengguna menekan tombol ‘Prediksi’ 5. Sistem melakukan peramalan sesuai dengan jumlah periode yang dimasukkan 6. Sistem menampilkan hasil peramalan pada jendela
<i>Alternate Flow Events</i>	<ol style="list-style-type: none"> 1. Pengguna telah berada dalam aplikasi peramalan, untuk melihat hasil <i>test error</i> dapat dengan menekan <i>tab</i> ‘Tabel Hasil Peramalan’
<i>Exception al Flow events</i>	<ul style="list-style-type: none"> • Komputer pengguna mati daya karena alasan tertentu, aplikasi akan tertutup secara otomatis

4.3.2.6. Use Case Menampilkan Grafik

Alur kegiatan untuk melihat grafik *data history* maupun hasil peramalan menggunakan periode yang ditentukan oleh pengguna dapat dilihat pada Tabel 4.6.

Tabel 4. 6. Use Case Menampilkan Grafik

<i>Use Case Name</i>	Menampilkan Grafik	
<i>Purpose</i>	Untuk melihat grafik data <i>history</i> atau hasil peramalan dengan model optimal	
<i>Actors</i>	Pengguna aplikasi	
<i>Pre-Condition</i>	<ul style="list-style-type: none"> • Pada komputer pengguna harus terpasang <i>Java Runtime Environment</i> (JRE) minimal versi 8.111 • Pengguna telah melakukan proses <i>training</i> dan proses <i>testing</i> untuk melihat grafik hasil ramalan 	
<i>Post-Condition</i>	Pengguna telah melihat hasil grafik data <i>history</i> dan/atau hasil ramalan	
<i>Flow events</i>	Melihat grafik data <i>history</i>	<ol style="list-style-type: none"> 1. Pengguna membuka Aplikasi Peramalan Produksi Ikan UPTD Pelabuhan Perikanan Banjarmains 2. Sistem menampilkan jendela mengenai data <i>history</i> 3. Pengguna menekan tombol ‘Grafik Data’ 4. Sistem menampilkan jendela grafik data <i>history</i>
	Melihat grafik hasil peramalan	<ol style="list-style-type: none"> 1. Pengguna menekan <i>tab</i> ‘Tabel Hasil Peramalan’ pada aplikasi 2. Sistem menampilkan jendela ‘Tabel Hasil Peramalan’ 3. Pengguna menekan tombol

		<p>‘Grafik’ pada bagian atas jendela</p> <p>4. Sistem menampilkan jendela baru mengenai grafik hasil periode yang dimasukkan</p>
<i>Alternate Flow Events</i>	<ul style="list-style-type: none"> • Pengguna telah berada dalam aplikasi peramalan, untuk melihat grafik <i>data history</i> dapat dengan menekan tombol ‘Grafik Data’ pada <i>tab</i> ‘Data Aktual’, sedangkan untuk melihat grafik hasil ramalan dengan menekan tombol ‘Grafik’ pada <i>tab</i> ‘Tabel Hasil Peramalan’ 	
<i>Exceptional Flow events</i>	<ul style="list-style-type: none"> • Komputer pengguna mati daya karena alasan tertentu, aplikasi akan tertutup secara otomatis 	

”Halaman ini sengaja dikosongkan”

BAB V IMPLEMENTASI

Bab ini merupakan pembahasan cara pengimplementasian dari tahapan perancangan sebelumnya yang telah dijelaskan pada Bab IV.

5.1. Pemasukan Data

Data yang digunakan sebagai masukan dalam penelitian ini adalah data jumlah produksi ikan UPTD Pelabuhan Perikanan Banjarmasin yang berjumlah 224 data sebagai variabel x dan rata-rata suhu udara pada stasiun Syamsudin Noor menurut NCEI (National Center for Environmental Information) yang berjumlah sama sebagai variabel y yang dibagi menjadi rasio 70% (156 periode) sebagai *data training* dan 30% (68 periode) sebagai *data testing*.

5.2. Pembentukan Model

Setiap model menggunakan fungsi *training* 'TrainGDX', yaitu proses *training* dengan menggunakan parameter-parameter yang ada. Parameter yang digunakan merupakan parameter *default* dari Matlab dan umumnya dijadikan sebagai acuan dengan peramalan dengan menggunakan metode BPNN. Berikut adalah parameter *default* yang dijelaskan:

***epoch* = 1000**

***lr* = 0.01**

***lr increase* = 1.05**

***lr decrease* = 0.7**

***mc* = 0.9**

***maxfail* = 6**

Keterangan:

epoch = jumlah iterasi

lr = *learning rate*

lr increase = penambahan *lr* apabila *error* membaik

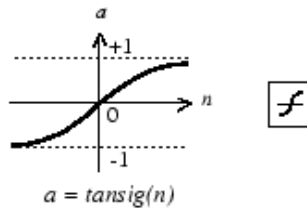
lr decrease = pengurangan *lr* apabila *error* memburuk

$mc = momentum$

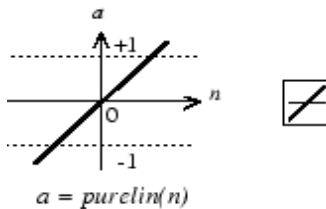
$maxfail =$ jumlah batas validasi *check*

Fungsi Aktivasi yang digunakan pada setiap model adalah fungsi '*tansig*' pada *hidden layer* dan fungsi '*purelin*' pada *output layer*, seperti pada Gambar 5.1. & 5.2.

Fungsi aktivasi *tansig* merubah nilai masukan menjadi keluaran bernilai -1, 0, atau 1, sedangkan fungsi aktivasi *purelin* menghasilkan nilai keluaran bernilai sama dengan nilai masukan [18].

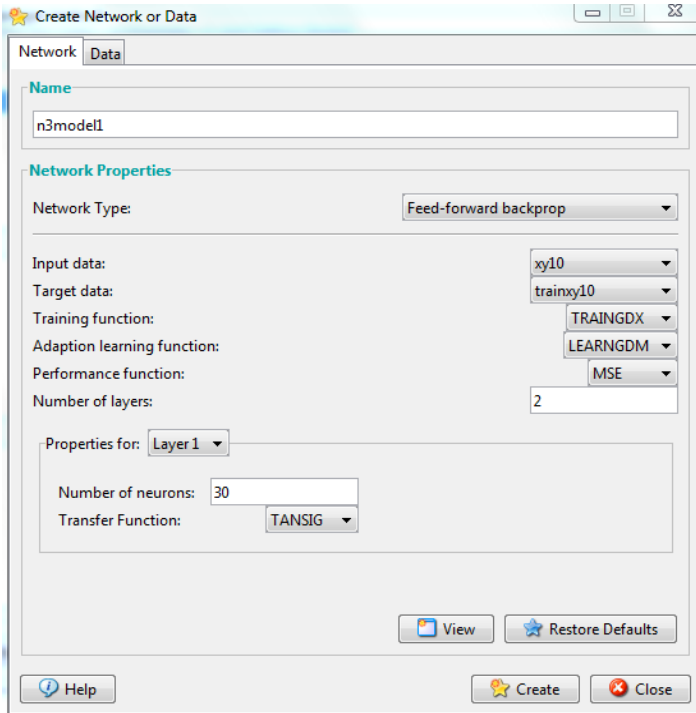


Gambar 5. 1. Fungsi Aktivasi *Tansig*



Gambar 5. 2. Fungsi Aktivasi *Purelin*

Pada Gambar 5.3. adalah contoh pembuatan model dengan nama '**n3model1**' menggunakan *input* xy10 (produksi ikan PPKB dari 1-5 periode sebelum target dan rerata suhu udara dari 1-5 periode sebelum target) dengan proses *training* menggunakan parameter (*TrainGDX*) dan pembaruan pembobotan untuk setiap iterasi (*LearnGDM*).



Gambar 5. 3. Contoh Pembuatan Model

Setiap model akan dilakukan proses pelatihan dan pengujian menggunakan neuron hidden layer dengan penentuan jumlah neuron hidden layer dengan formula m , $2m$, dan $3m$.

Keterangan:

m = jumlah *neuron* masukan

5.2.1. Proses *Training*

Proses *training* menggunakan data set *training* yang telah dibagi sebelumnya, yaitu 70% rasio data *training* yang berjumlah 156 periode dan 30% data *testing* yang berjumlah 68 periode.

Pada setiap model, proses *training* dilakukan sebanyak 3 kali yang bertujuan untuk menyesuaikan penentuan bobot awal yang selalu berubah-ubah dalam menentukan kesalahan minimal.

Fungsi yang digunakan adalah *adaptive learning function*, yaitu proses pembelajaran dengan penyesuaian parameter yang digunakan dan pembaruan pembobotan pada setiap iterasi. Sedangkan fungsi aktivasinya adalah fungsi aktivasi ***tansig*** pada *hidden layer* dan fungsi aktivasi ***purelin*** pada *output layer* yang telah dijelaskan pada sub-bab sebelumnya, seperti pada Gambar 5.4.

Berikut adalah parameter yang digunakan yang telah dijelaskan pada sub-bab sebelumnya:

***epoch* = 1000**
***lr* = 0.01**
***lr increase* = 1.05**
***lr decrease* = 0.7**
***mc* = 0.9**
***maxfail* = 6**

Keterangan:

epoch = jumlah iterasi

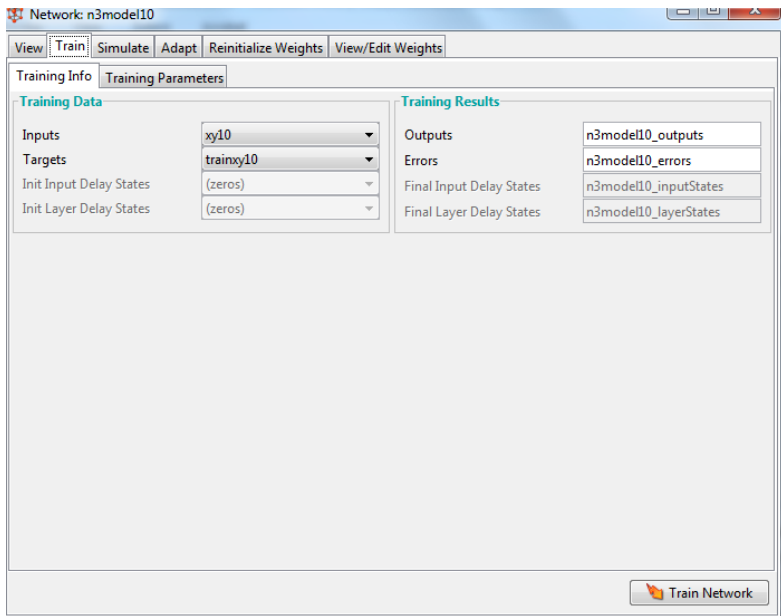
lr = *learning rate*

lr increase = penambahan *lr* apabila *error* membaik

lr decrease = pengurangan *lr* apabila *error* memburuk

mc = *momentum*

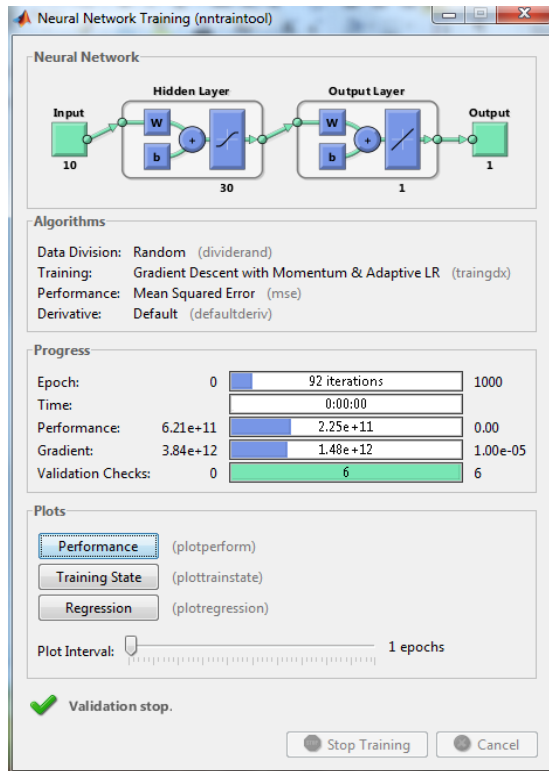
maxfail = jumlah batas validasi *check*



Gambar 5. 4. Contoh Proses *Training*

Gambar 5.4. menunjukkan proses *training* dengan menggunakan 'nntool' pada Matlab, sebagai *input* adalah xy10, yaitu data *training* jumlah produksi ikan PPKB 1-5 periode sebelumnya dan rata-rata suhu udara dari 1-5 periode sebelumnya.

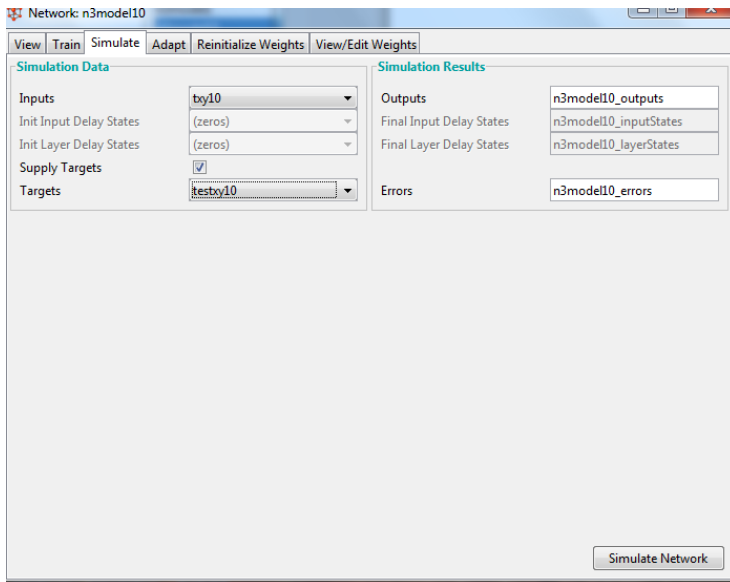
Sedangkan pada Gambar 5.5. merupakan contoh hasil *training* yang umumnya digunakan untuk melihat performa proses *training*.



Gambar 5. 5. Contoh Hasil *Training*

5.2.2. Proses *Testing*

Setelah didapatkan model dari hasil pembobotan *proses training*, kemudian dilanjutkan proses *testing* menggunakan data *testing*. Hal ini bertujuan untuk melihat performa model terhadap data yang berbeda.



Gambar 5. 6. Contoh Proses *Testing*

Gambar 5.6. menunjukkan proses *testing* dengan menggunakan 'nntool' pada Matlab, sebagai *input* adalah txy10, yaitu data *testing* jumlah produksi ikan PPKB 1-5 periode sebelumnya dan rata-rata suhu udara dari 1-5 periode sebelumnya.

5.3. *Forecast*

Setelah didapatkan model optimal dari analisa model melalui hasil *error* (MAPE) yang didapatkan, proses peramalan dilakukan dengan menggunakan seluruh *data history*, yaitu data dari periode Januari 1998 – Agustus 2016. Hal ini bertujuan untuk menarik kesimpulan akhir apakah metode BPNN baik digunakan untuk peramalan jumlah produksi ikan seperti *dataset* Pelabuhan Perikanan Banjarmasin ini ataukah tidak. Dan kemudian dilakukan peramalan untuk satu periode kedepan dengan menggunakan model optimal.

Peramalan dilakukan dengan menggunakan model 3 (model optimal) dengan jumlah neuron *hidden layer* 2 dengan parameter *default* pada Matlab.

5.4. Implementasi Aplikasi

Pada sub-bab ini akan dijelaskan mengenai proses implementasi dari hasil perancangan aplikasi yang telah dijelaskan pada Bab IV.

5.4.1. Fungsi Menampilkan Data History

Syntax yang digunakan untuk menampilkan hasil peramalan pada jendela GUI untuk kebutuhan fungsi Menampilkan Data History dijelaskan pada *Script 5.1*.

```
Scanner scanner = new Scanner(new File("Produksi_ikan.csv"));
scanner.useDelimiter(",");

int i = tabel.getRowCount()+1;
int col = 0;
while(scanner.hasNext()){
    String isi = (String) scanner.next();
    dataProdIkan[i-1][col] = isi;

    if(i > tabel.getRowCount()){
        DefaultTableModel model = (DefaultTableModel) tabel.getModel();
        model.addRow(new Object[]{"", ""});
    }

    if(col == 0){
        tabel.setValueAt(isi, i-1, 0);
        col = 1;
    }else if(col == 1){
        tabel.setValueAt(isi, i-1, 1);
        col = 2;
    }else if(col == 2){
        tabel.setValueAt(isi, i-1, 2);
        col = 0;
        i++;
    }
}
```

Script 5. 1. Fungsi Menampilkan Data History

Pada *Script 5.1.*, dijelaskan bahwa hal pertama yang dilakukan adalah membuat tabel untuk menampilkan data aktual dengan nama 'tabel' pada JPanel, kemudian memasukkan perintah membaca file 'Produksi_ikan.csv' dengan *syntax* 'scanner', setelah itu dilakukan penginputan data seperti perulangan di atas.

5.4.2. Fungsi Training

Syntax yang digunakan untuk melakukan proses *training* untuk kebutuhan fungsi Training bagi pengguna aplikasi dijelaskan pada *Script 5.2.*

```
NeuralNetwork neuralNetwork = new MultiLayerPerceptron(2,2,1);
DataSet trainingSet = new DataSet(2, 1);

for(int i = 0; i < trainProdIkan; i++){
    double[] x = new double[] {
        Integer.parseInt(inputProdIkan[i][1])/d,
        Integer.parseInt(inputProdIkan[i+1][1])/d
    };
    double[] y = new double[] {Integer.parseInt(inputProdIkan[i+2][1])/d};
    trainingSet.addRow(new DataSetRow (x, y));
}

BackPropagation bp = new BackPropagation();
bp.setMaxIterations(1000);
bp.setMaxError(0.0005);
bp.setLearningRate(0.01);

neuralNetwork.learn(trainingSet, bp);
```

Script 5. 2. Fungsi Training

Pada *Script 5.2.*, dijelaskan bahwa hal pertama yang dilakukan adalah membuat model dengan arsitektur 2 *neuron input*, 2 *neuron hidden layer*, dan 1 *neuron output* dan pembuatan *dataset* dengan 2 data masukan untuk 1 periode target, dengan data masukan adalah 1 dan 2 periode sebelum target.

Setelah itu dilakukan pemanggilan fungsi *backpropagation* dan pendefinisian parameter yang akan digunakan, yaitu jumlah iterasi sebanyak 1000, jumlah ambang batas *error*

sejumlah 0.0005, dan nilai *learning rate* sejumlah 0,01, dan terakhir adalah melakukan proses *training* dengan *syntax* 'neuralnetwork.learn'.

5.4.3. Fungsi Testing

Syntax yang digunakan untuk pembuatan kebutuhan fungsi *testing* pada aplikasi pembantu peramalan dijelaskan pada *Script 5.3*.

```
public void percobaan(){
    NeuralNetwork neuralNetwork = NeuralNetwork.createFromFile("or_perceptron.nnet");

    for(int i = (int)trainProdIkan; i < jumlahData-2; i++){
        neuralNetwork.setInput(
            Integer.parseInt(inputProdIkan[i][1])/d,
            Integer.parseInt(inputProdIkan[i+1][1])/d
        );
        neuralNetwork.calculate();
        double[] hasil = neuralNetwork.getOutput();
        double aktual = Double.parseDouble(inputProdIkan[i+2][1]);
        double APE = Math.abs(aktual - Math.round(hasil[0]*d))/aktual;
        TAPE += APE;
    }
}
```

Script 5. 3. Fungsi Testing

Pada *Script 5.3.*, dijelaskan bahwa hal pertama yang dilakukan adalah pendefinisian *input* dan *target*. *Input* adalah data 1 periode dan 2 periode sebelum *target*, kemudian dilakukan pendefinisian variabel untuk perulangan iterasi, selama jumlah *data testing* yang digunakan belum mencapai baris 224, maka penginputan terus dilakukan. Setiap data yang diinput dilakukan *testing* dengan *syntax* 'neuralNetwork.calculate()'. Hasil *output* dari proses *testing* kemudian dibandingkan dengan data target untuk dilihat *errornya*.

5.4.4. Fungsi Menampilkan Testing Error

Syntax yang digunakan untuk membuat kebutuhan fungsi Menampilkan Testing Error, yaitu fungsi agar pengguna

dapat melihat perbedaan dari hasil data *output* dan data target dijelaskan pada *Script 5.4*.

```
public double getMAPE() {
    return (TAPE/testProdIkan)*100;
}
```

Script 5. 4. Fungsi Menampilkan Testing Error

Pada *Script 5.4.*, dijelaskan bahwa hasil penjumlahan dari variabel APE pada proses *testing* dibagi dengan jumlah data *testing* (64 periode) dan dikali 100%.

5.4.5. Fungsi Forecast

Syntax untuk membuat kebutuhan fungsi Forecast, yaitu kebutuhan agar pengguna aplikasi dapat melakukan peramalan menggunakan *dataset* dijelaskan pada *Script 5.4.* dan *Script 5.5*.

```
public double[] peramalan(int periode){
    NeuralNetwork neuralNetwork = NeuralNetwork.createFromFile("or_perceptron.nnnet");
    double[] networkOutput = new double[periode+2];
    networkOutput[0] = Integer.parseInt(inputProdIkan[jumlahData-2][1]);
    networkOutput[1] = Integer.parseInt(inputProdIkan[jumlahData-1][1]);

    for(int i = 0; i < periode; i++){
        neuralNetwork.setInput (
            networkOutput[i]/d,
            networkOutput[i+1]/d
        );
        neuralNetwork.calculate();
        double[] hasil = neuralNetwork.getOutput();
        networkOutput[i+2] = Math.round(hasil[0]*d);
    }

    return networkOutput;
}
```

Script 5. 5. Script 1 Fungsi Forecast

Pada *Script 5.5.*, dijelaskan bahwa hal pertama yang dilakukan adalah pendefinisian data *input*, yaitu data 1 periode dan 2 periode sebelum variabel data hasil (*output*), kemudian dilakukan perulangan hingga mencapai batas periode yang dimasukkan oleh pengguna.

Pada *Script* 5.6. dijelaskan bagaimana alur proses penginputan hasil prediksi pada tabel. Yaitu perulangan penginputan variabel 'hasilRamal' hingga jumlah periode tercapai.

```
int periode = Integer.parseInt(periodeRamal2.getText());
double[] hasilRamal = nn.peramalan(periode);

for (int i = 0; i < periode ; i++) {
    if(i > tabel2.getRowCount()-1){
        DefaultTableModel model = (DefaultTableModel) tabel2.getModel();
        model.addRow(new Object[]{"", "", "", ""});
    }

    tabel2.setValueAt(i+1,i,0);
    tabel2.setValueAt(hasilRamal[i+2],i,1);
    tabel2.setValueAt(hasilRamal[i+2]-hasilRamal[i+1],i,2);
    if(hasilRamal[i+2] > hasilRamal[i+1]){
        tabel2.setValueAt("Naik",i,3);
    }else if(hasilRamal[i+2] < hasilRamal[i+1]){
        tabel2.setValueAt("Turun",i,3);
    }else{
        tabel2.setValueAt("Tetap",i,3);
    }
}
```

Script 5. 6. Script 2 Fungsi Forecast

5.4.6. Fungsi Menampilkan Grafik

Syntax yang digunakan untuk membuat kebutuhan fungsi Menampilkan Grafik *data history* pada aplikasi dijelaskan pada *Script* 5.7.

```
List<Integer> scores = new ArrayList<Integer>();
for (int i = 0; i < jumlahData-2 ; i++) {
    String angka = dataProdIkan[i][1];
    scores.add(Integer.parseInt(angka));
}

DrawGraph mainPanel = new DrawGraph(scores);

JFrame frame = new JFrame("Grafik Data Historis Produksi Ikan");
frame.getContentPane().add(mainPanel);
frame.pack();
frame.setVisible(true);
frame.setLocation(75, 75);
```

Script 5. 7. Script 1 Fungsi Menampilkan Grafik

Pada *Script 5.7.*, hal pertama yang dilakukan adalah pendefinisian variabel 'scores', kemudian dilakukan penginputan nilai variabel 'dataProdIkan' (jumlah produksi ikanPPKB) pada variabel 'scores' hingga batas akhir iterasi (224 iterasi).

Pada *Script 5.8.* dijelaskan *syntax* yang digunakan dalam pembuatan kebutuhan fungsi menampilkan data hasil ramalan pada aplikasi.

```
int periode = Integer.parseInt(periodeRamal2.getText());
double[] hasilRamal = nn.peramalan(periode);

List<Integer> scores = new ArrayList<Integer>();
for (int i = 0; i < periode ; i++) {
    Double d = hasilRamal[i+2];
    Integer in = d.intValue();
    scores.add(in);
}
DrawGraph mainPanel = new DrawGraph(scores,5);

JFrame frame = new JFrame("Grafik Data Hasil Peramalan Produksi Ikan");
frame.getContentPane().add(mainPanel);
frame.pack();
frame.setVisible(true);
frame.setLocation(75, 75);
```

Script 5. 8. Script 2 Fungsi Menampilkan Grafik

Pada Gambar 5.14., dijelaskan bahwa hal awal yang dilakukan adalah pendefinisian variabel 'hasilRamal', yaitu variabel yang diambil dari hasil *output* proses peramalan, kemudian dilakukan iterasi pemasukan data kepada variabel 'scores' dengan menggunakan nilai hasil ramalan tersebut hingga batas periode yang dimasukkan sebelumnya oleh pengguna. Hal terakhir adalah penggambaran grafik dengan *syntax* 'DrawGraph mainPanel'.

”Halaman ini sengaja dikosongkan”

BAB VI HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas hasil dari proses implementasi yang dibahas pada bab sebelumnya, yaitu pada Bab Implementasi dan akan dilakukan analisa dari hasil implementasi tersebut.

6.1. Hasil Implementasi Model

Sub-bab ini menjelaskan mengenai hasil dari implementasi proses *training* dan *testing* sebanyak 3 kali dengan menggunakan kesepuluh model dengan jumlah *neuron hidden layer* n , $2n$, dan $3n$ dan parameter yang telah dijelaskan pada bab sebelumnya, yaitu:

***epoch* = 1000**

***lr* = 0.01**

***lr increase* = 1.05**

***lr decrease* = 0.7**

***mc* = 0.9**

***maxfail* = 6**

Keterangan:

epoch = jumlah iterasi

lr = *learning rate*

lr increase = penambahan *lr* apabila *error* membaik

lr decrease = pengurangan *lr* apabila *error* memburuk

mc = *momentum*

maxfail = jumlah batas validasi *check*

6.1.1. Hasil Train & Test Error (n hidden layer)

Pada sub-bab ini akan dijelaskan mengenai hasil proses *training* dan proses *testing* menggunakan kesepuluh model dengan menggunakan jumlah *neuron hidden layer* sebanyak n -input.

Keterangan:

n = jumlah *neuron* masukan

Tabel 6.1. merupakan penggambaran perbandingan *MAPE error* yang dihasilkan oleh setiap model dengan menggunakan jumlah *neuron hidden layer* sebanyak *n-input* dengan menggunakan tabel.

Pada Tabel 6.1. didapatkan *error* terendah untuk setiap model melalui tiga kali proses *training* dan *testing* dengan rincian:

- Model 1 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-2 dengan 24.5393% *error*
- Model 2 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-2 dengan 28.6436% *error*
- Model 3 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-3 dengan 23.1304% *error*
- Model 4 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-1 dengan 24.6907% *error*
- Model 5 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-3 dengan 23.9022% *error*
- Model 6 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-2 dengan 27.0468 *error*
- Model 7 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-2 dengan 27.0512% *error*
- Model 8 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-3 dengan 43.1282% *error*

- Model 9 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-2 dengan 24.81268% *error*
- Model 10 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-3 dengan 31.7529% *error*

Didapatkan model optimal (model dengan *min. error* terendah) untuk *m neuron hidden* layer adalah model 3, yaitu model dengan *neuron* masukan (*x-1*) dan (*x-2*) dengan *error* 23.1304 yang berdasarkan pada Tabel 6.1

Keterangan:

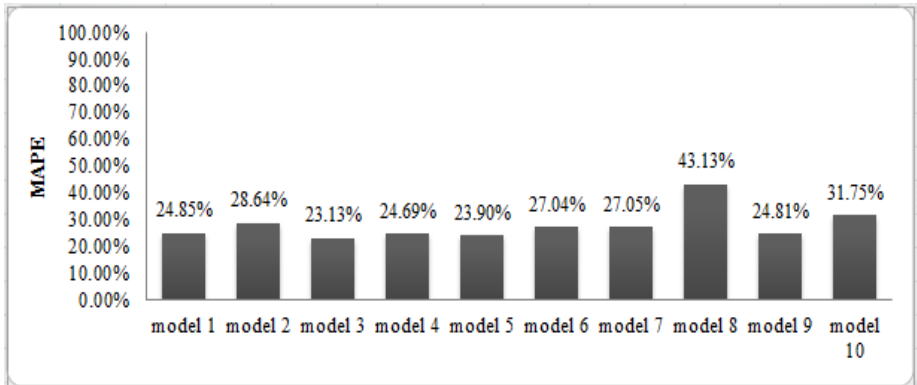
m = jumlah *neuron* masukan

x - n = jumlah produksi ikan pada *n* periode sebelumnya

Tabel 6. 1. Tabel Perbandingan *Error (n hidden layer)*

Model	MAPE					
	Training			Testing		
	Train 1	Train 2	Train 3	Test 1	Test 2	Test 3
1	17.6964%	18.6515%	18.6512%	24.8848%	24.5393%	24.5394%
2	20.6390%	24.5587%	19.5426%	31.3635%	28.6436%	32.6093%
3	22.6307%	19.8293%	20.8003%	28.0854%	24.4519%	23.1304%
4	19.1348%	18.7480%	18.7479%	24.6907%	24.7190%	24.7190%
5	19.0626%	17.9147%	19.9407%	23.9447%	24.2749%	23.9022%
6	19.5096%	18.5556%	20.7455%	27.8521%	27.0468%	27.5507%
7	19.8453%	18.6766%	18.6770%	27.9315%	27.0512%	27.0514%
8	27.5896%	26.1250%	23.6394%	47.1447%	45.3176%	43.1282%
9	19.9741%	18.5796%	18.5794%	34.7683%	24.81268%	24.81270%
10	28.1097%	28.1095%	24.9573%	38.8440%	38.8443%	31.7529%

Gambar 6.1. merupakan penggambaran perbandingan MAPE *error* antar-model n *neuron hidden layer* dengan menggunakan grafik dengan x -axis adalah nama model dan y -axis adalah persen MAPE.



Gambar 6. 1. Grafik Perbandingan *Error* (n *hidden layer*)

6.1.2. Hasil *Train & Test Error* ($2n$ *hidden layer*)

Pada sub-bab ini dijelaskan mengenai hasil proses *training* dan proses *testing* sebanyak tiga kali menggunakan kesepuluh model dengan menggunakan jumlah *neuron hidden layer* sebanyak $2n$ -*input*.

Keterangan:

n = jumlah *neuron* masukan

Tabel 6.2. merupakan penggambaran perbandingan MAPE *error* yang dihasilkan oleh setiap model dengan menggunakan jumlah *neuron hidden layer* sebanyak $2n$ -*input* dengan menggunakan tabel.

Pada Tabel 6.2. didapatkan *error* terendah untuk setiap model melalui tiga kali proses *training* dan *testing* dengan rincian:

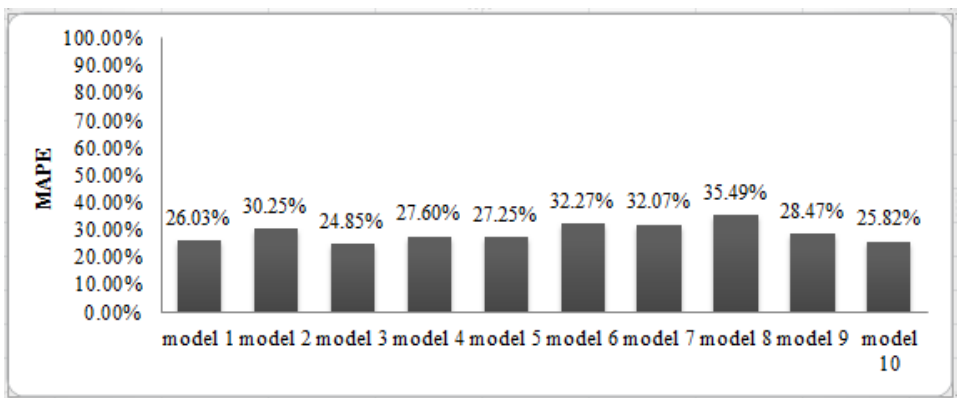
- Model 1 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-3 dengan 26.0314% *error*
- Model 2 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-3 dengan 30.2478% *error*
- Model 3 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-1 dengan 24.8484% *error*
- Model 4 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-3 dengan 27.5986% *error*
- Model 5 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-1 dengan 27.2543% *error*
- Model 6 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-3 dengan 32.2707 *error*
- Model 7 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-3 dengan 32.0658% *error*
- Model 8 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-3 dengan 35.4885% *error*
- Model 9 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-2 dengan 28.4734% *error*

- Model 10 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-3 dengan 25.8170% *error*

Tabel 6. 2. Tabel Perbandingan Error (2n hidden layer)

Model	MAPE					
	Training			Testing		
	Train 1	Train 2	Train 3	Test 1	Test 2	Test 3
1	33.1449%	21.3488%	19.8746%	53.1817%	26.8232%	26.0314%
2	7.0657%	7.0657%	6.8085%	30.4505%	30.4506%	30.2478%
3	19.1039%	17.7658%	18.0710%	24.8484%	25.7067%	25.4309%
4	19.1362%	19.4922%	19.4922%	28.6767%	27.5987%	27.5986%
5	18.7053%	18.7054%	18.7054%	27.2543%	27.2544%	27.2544%
6	17.9689%	17.9694%	17.3299%	32.6001%	32.6001%	32.2707%
7	31.2564%	31.2576%	22.4089%	44.8029%	44.8028%	32.0658%
8	41.5615%	25.6481%	24.7802%	44.0284%	36.0068%	35.4885%
9	22.9609%	20.8518%	20.8517%	36.9432%	28.4736%	28.4734%
10	18.2149%	18.2149%	18.2149%	25.8170%	25.8171%	25.8171%

Gambar 6.2. merupakan penggambaran perbandingan MAPE *error* antar-model 2n neuron hidden layer dengan menggunakan grafik dengan *x-axis* adalah nama model dan *y-axis* adalah persen MAPE.



Gambar 6. 2. Grafik Perbandingan Error (2n hidden layer)

Berdasarkan Tabel 6.2. dan Gambar 6.2. didapatkan model optimal (model dengan *min. error* terendah) untuk $2m$ *neuron hidden layer* adalah model 3, yaitu model dengan *neuron* masukan ($x-1$) dan ($x-2$) dengan *error* 24.8484%.

Keterangan:

m = jumlah *neuron* masukan

$x - n$ = jumlah produksi ikan pada n periode sebelumnya

6.1.3. Hasil Train & Test Error (3n hidden layer)

Pada sub-bab ini dijelaskan mengenai hasil proses *training* dan proses *testing* menggunakan kesepuluh model dengan menggunakan jumlah *neuron hidden layer* sebanyak $3n$ -*input*.

Keterangan:

n = jumlah *neuron* masukan

Tabel 6.3. merupakan penggambaran perbandingan MAPE *error* yang dihasilkan oleh setiap model dengan menggunakan jumlah *neuron hidden layer* sebanyak $3n$ -*input* dengan menggunakan tabel.

Pada Tabel 6.3. didapatkan *error* terendah untuk setiap model melalui tiga kali proses *training* dan *testing* dengan rincian:

- Model 1 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-1 dengan 24.8516% *error*
- Model 2 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-2 dengan 26.0678% *error*

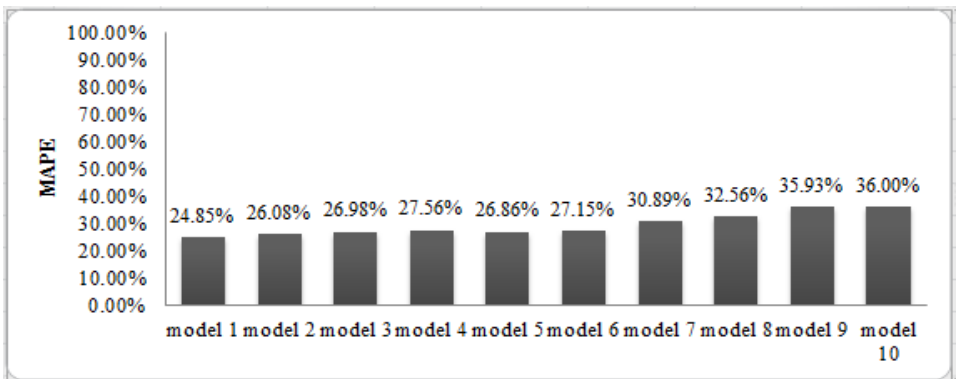
- Model 3 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-3 dengan 26.98313% *error*
- Model 4 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-2 dengan 27.5647% *error*
- Model 5 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-1 dengan 26.8597% *error*
- Model 6 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-3 dengan 27.1458% *error*
- Model 7 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-3 dengan 30.8804% *error*

Tabel 6. 3. Tabel Perbandingan *Error* (3n hidden layer)

Model	MAPE					
	Training			Testing		
	Train 1	Train 2	Train 3	Test 1	Test 2	Test 3
1	22.8316%	19.0321%	19.4112%	24.8516%	26.1859%	25.6208%
2	21.2572%	20.0805%	21.8703%	26.4842%	26.0788%	24.7372%
3	18.9765%	18.9764%	18.8513%	28.2134%	28.2135%	26.9813%
4	17.7299%	20.5755%	18.0218%	27.7718%	27.5647%	27.6768%
5	18.3812%	18.3812%	17.6898%	27.6427%	27.6428%	26.8597%
6	18.6414%	18.6410%	18.6410%	27.1458%	27.1459%	27.1459%
7	18.9869%	18.9870%	18.3207%	34.4890%	34.4891%	30.8804%
8	17.6337%	17.6338%	17.6338%	32.5650%	32.5651%	32.5651%
9	20.1196%	20.1197%	20.6034%	36.5837%	36.5836%	35.9295%
10	29.4139%	19.7531%	18.7591%	56.8385%	36.2041%	36.0049%

- Model 8 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-3 dengan 32.5650% *error*
- Model 9 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-2 dengan 36.5836% *error*
- Model 10 mendapatkan hasil optimal (*min. error*) pada pembobotan hasil *training* dan proses *testing* ke-3 dengan 36.2041% *error*

Gambar 6.3. merupakan penggambaran perbandingan MAPE *error* antar-model *3n-input neuron hidden layer* dengan menggunakan grafik dengan *x-axis* adalah nama model dan *y-axis* adalah persen MAPE.



Gambar 6. 3 Grafik Perbandingan Error (*3n hidden layer*)

Berdasarkan Tabel 6.3. dan Gambar 6.3. didapatkan model optimal (model dengan *min. error* terendah) untuk *3m neuron hidden layer* adalah model 1, yaitu model dengan *neuron* masukan ($x-1$) dengan *error* 24.8516%.

Keterangan:

m = jumlah *neuron* masukan

$x - n$ = jumlah produksi ikan pada n periode sebelumnya

6.2. Analisa Hasil Implementasi

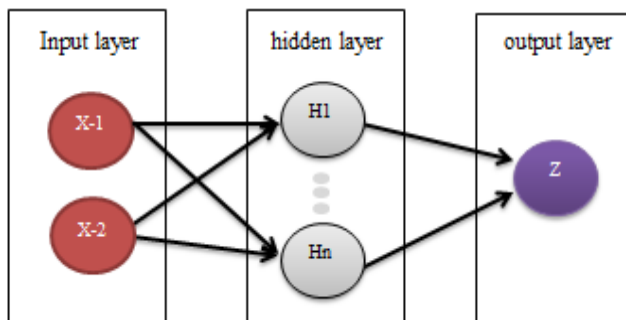
Sub-bab ini akan menjelaskan mengenai hasil analisa berdasarkan hasil implementasi *dataset* jumlah produksi ikan PPKB dan rata-rata suhu udara Stasiun Syamsudin Noor yang berjumlah 224 periode dengan menggunakan jumlah *neuron hidden layer* (n , $2n$, dan $3n$) sebanyak tiga kali proses *training* dan *testing* dengan menggunakan parameter yang telah ditentukan.

Keterangan:

n = jumlah *neuron* masukan

6.2.1. Analisa Model

Berdasarkan hasil implementasi kesepuluh model seperti dijelaskan pada sub-bab 6.1., maka disimpulkan bahwa model optimal (model dengan *minimum error* terendah) adalah model 3, yaitu model dengan *input* $x-1$ dan $x-2$ (lihat Gambar 6.4.) dengan jumlah *neuron hidden layer* adalah n , yaitu 2 *neuron hidden layer*. Model 3 menghasilkan *min. error* terendah dengan 23.1304% *error* (lihat Tabel 6.1.) dibandingkan dengan kesembilan model lainnya pada penggunaan jumlah *neuron hidden layer* sebanyak n , $2n$, maupun $3n$.



Gambar 6. 4. Model Optimal

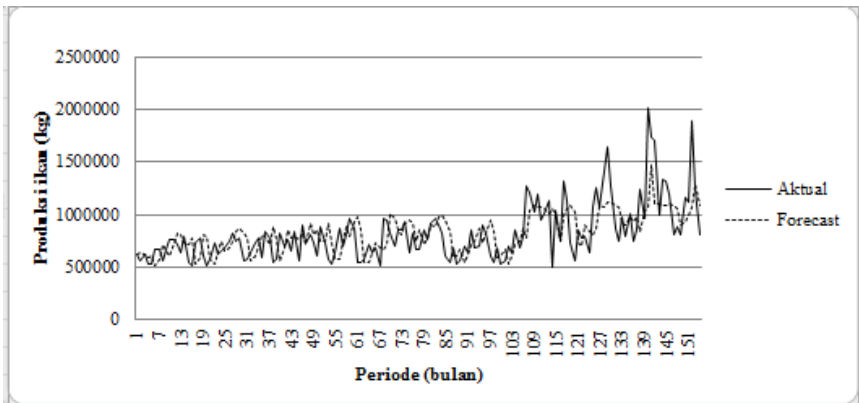
Keterangan:

x = *neuron* masukan (jumlah produksi ikan)

h = *neuron hidden layer*

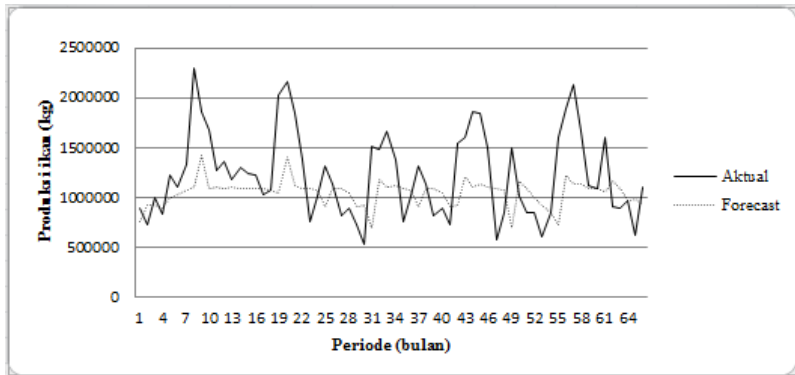
z = *neuron output*

Pada Gambar 6.5. dan Gambar 6.6 adalah gambaran hasil proses *training* dan proses *testing* model 3 menggunakan *dataset* yang dibagi menjadi 70% data *train set* (156 periode) dan 30% data *test set* (68 periode).



Gambar 6. 5. Hasil *Training* Model 3 (n hidden layer)

Pada Gambar 6.5. dilihat sekilas bahwa hasil proses *training* jumlah produksi ikan PPKB dengan model 3 mendekati dengan nilai data *target* (aktual) dari periode awal hingga periode 124, pada periode 125 dan setelahnya terjadi kenaikan dan perubahan pola pada data aktual dan dan dinilai proses *training* dengan menggunakan model 3 tidak dapat mengikuti perubahan pola ini.



Gambar 6. 6. Hasil *Testing Model 3* (n hidden layer)

Pada Gambar 6.6. dilihat sekilas bahwa hasil *testing dataset* pada 64 periode dengan menggunakan model 3 agak menjauhi dari nilai data sebenarnya (aktual), hal ini diduga karena disebabkan oleh tidak dapatnya model 3 mengikuti kenaikan dan perubahan pola pada periode 125 hingga 154 pada proses *training* (lihat Gambar 6.5.)

6.2.2. Analisa Variabel Pendukung (Suhu Udara)

Berdasarkan hasil implementasi seluruh model menggunakan tiga jumlah *neuron hidden layer* yang berbeda (m , $2m$, dan $3m$) dengan tiga kali proses *training* dan *testing* (lihat Tabel 6.4), maka dapat ditarik kesimpulan bahwa variabel suhu udara kurang mempengaruhi terhadap perhitungan peramalan dengan menggunakan metode BPNN untuk *dataset* seperti jumlah produksi ikan pada UPTD Pelabuhan Perikanan Banjarmasin ini dengan melihat dari perbedaan keluaran MAPE yang tidak jauh berbeda dari model yang menggunakan variabel pendukung dan tidak menggunakan variabel pendukung, yaitu dengan perbedaan rata-rata hasil *error* proses *training* sejumlah 0.12% dan rata-rata hasil *error* proses *testing* sejumlah 3.08%. Perbedaan hasil keluaran ini digunakan untuk menarik kesimpulan bahwa variabel pendukung (suhu udara) kurang

mempengaruhi terhadap hasil akhir penelitian untuk studi kasus ini (lihat sub-bab 2.2.5.).

Keterangan:

m = jumlah *neuron* masukan

$x - n$ = jumlah produksi ikan pada n periode sebelumnya

Tabel 6. 4. Performa *Input* Variabel Pendukung

Model ANN		MAPE Minimum Neuron Hidden Layer						Average MAPE
		N		2N		3N		
		train	test	train	test	train	test	
Tanpa Variabel Pendukung	Model 1	17.69%	24.54%	19.87%	26.03%	19.03%	24.85%	Train = 19.09% Test = 27.23%
	Model 3	19.83%	23.13%	17.77%	24.85%	18.85%	26.98%	
	Model 5	17.91%	23.90%	18.70%	27.25%	17.70%	27.64%	
	Model 7	18.67%	27.05%	22.41%	32.07%	18.32%	30.88%	
	Model 9	18.58%	24.81%	20.85%	28.47%	20.12%	35.93%	
Menggunakan Variabel Pendukung	Model 2	19.54%	28.64%	6.81%	30.25%	20.08%	24.74%	Train = 18.97% Test = 30.31%
	Model 4	18.75%	24.69%	19.14%	27.60%	17.73%	27.56%	
	Model 6	18.56%	27.05%	17.33%	32.27%	18.64%	27.15%	
	Model 8	23.64%	43.13%	24.78%	35.49%	17.63%	32.56%	
	Model 10	24.96%	31.75%	18.21%	25.82%	18.76%	36.00%	

6.2.3. Analisa Parameter

Pada sub-bab ini akan dijelaskan mengenai hasil analisa terhadap penggunaan variasi parameter terhadap model optimal (model 3), yaitu model dengan *input* (x-1) dan (x-2) dengan 2 *neuron hidden layer*.

6.2.3.1. Learning Rate

Analisa parameter *learning rate* ini dilakukan dengan didasari pertanyaan, apakah semakin lama proses *training* akan menghasilkan hasil *error* yang semakin kecil (semakin rendah *learning rate*, proses *training* menjadi semakin lama).

Pada implementasi digunakan model 3 (model optimum), yaitu model dengan *input* (x-1) dan (x-2) dengan 2 *neuron hidden layer* untuk menganalisa penggunaan parameter

learning rate bagi hasil akhir peramalan untuk kasus ini. Hasil implementasi variasi parameter *learning rate* menggunakan model 3 dijelaskan pada Tabel 6.5.

Tabel 6. 5. Performa Variasi *Learning Rate*

Model 3 (x-1 dan y-1) 2 neuron hidden layer								
Learning Rate	Training MAPE				Testing MAPE			
	Train 1	Train 2	Train 3	average MAPE	Test 1	Test 2	Test 3	average MAPE
0.01 (default)	22.6307%	19.8293%	20.8003%	21.0868%	28.0854%	24.4519%	23.1304%	25.2106%
0.1	24.3905%	20.0797%	20.0795%	21.5165%	31.3770%	26.5950%	26.5950%	28.1890%
0.2	20.0795%	20.0795%	21.7079%	20.6223%	26.5950%	26.5950%	26.3053%	26.4984%
0.3	19.2817%	19.5685%	19.5685%	19.4729%	26.5063%	26.3131%	26.3132%	26.3776%
0.4	21.1444%	21.1452%	21.8896%	21.3931%	26.0804%	26.0805%	26.1030%	26.0880%
0.5	20.4905%	20.4899%	20.4899%	20.4901%	25.8505%	25.8505%	25.8505%	25.8505%
0.6	19.2237%	18.9743%	19.6132%	19.2704%	26.0077%	26.0882%	25.8869%	25.9943%
0.7	18.8869%	22.0167%	22.0165%	20.9734%	26.4062%	26.0012%	26.0013%	26.1362%
0.8	19.0846%	20.9832%	19.1156%	19.7278%	26.2669%	25.9181%	26.1901%	26.1250%
0.9	20.4869%	19.2449%	19.2449%	19.6589%	25.8890%	26.0526%	26.0526%	25.9981%
(average max) – (average min)				2.2461%	all test average			26.2468%

Dilihat pada Tabel 6.5., penggunaan *learning rate* yang berbeda-beda dinilai kurang mempengaruhi terhadap hasil akhir peramalan, yaitu dilihat dari nilai *train error* yang dihasilkan (*learning rate* hanya digunakan pada proses *training*), yaitu kecilnya perbedaan antara nilai rata-rata *error* tertinggi dan rata-rata *error* terendah yang hanya mencapai 2.25%.

Dilihat pada Tabel 6.5., dapat dijawab pertanyaan yang mendasari analisa ini, apakah semakin lama proses *training* akan menghasilkan nilai *error* yang semakin rendah pula (semakin lama proses *training*, nilai *learning rate* semakin rendah). Pada Tabel 6.5. dapat dilihat bahwa, pada *dataset* ini semakin lama proses *training* tidak menghasilkan nilai *error* yang semakin rendah (*learning*

rate 0.1, memiliki hasil rata-rata *error* lebih tinggi dibandingkan *learning rate* 0.2). Di sisi lain, hasil *train error* yang rendah tidak semata-mata menghasilkan hasil *test error* yang rendah pula (pada *learning rate* 0.7 menghasilkan *train error* terendah (terbaik), akan tetapi tidak menghasilkan *test error* terbaik pula).

Berdasarkan analisa dari sudut pandang pertama, penggunaan variasi *learning rate* kurang mempengaruhi terhadap hasil akhir penelitian, yaitu dilihat dari kecilnya perbedaan antara rata-rata *train error* tertinggi dan terendah hanya 2.25% (lihat sub-bab 2.2.5.) dan hampir semua variasi *learning rate* menghasilkan *test error* mendekati 26.25%, kecuali *learning rate* 0.1 pada *test* 1 yang menghasilkan perbedaan > 5% dengan perbandingan terjadinya hal ini adalah 1:30 percobaan (*test*).

Dari sudut pandang kedua, penentuan *learning rate* optimal tidak semata-mata dapat dilihat dari satu atau dua aspek aja, contohnya semakin tinggi atau semakin rendah nilai *learning rate* akan menghasilkan *error* yang semakin tinggi atau semakin rendah pula. Jadi kesimpulan dari sudut pandang ini adalah untuk mendapatkan parameter *learning rate* optimal dapat dilakukan proses *trial & error*, yaitu mencoba sebanyak-banyaknya variasi nilai parameter *learning rate* yang berbeda agar mendapatkan nilai *test error* yang bervariasi pula yang nantinya dibandingkan yang mana merupakan nilai parameter *learning rate* optimal.

Umumnya digunakan nilai *learning rate* sejumlah 0.01 untuk peramalan menggunakan metode BPNN sesuai acuan pada Matlab.

6.2.3.2. Epoch

Hampir sama seperti analisa pada sub-bab sebelumnya, pertanyaan yang mendasari analisa pada parameter ini adalah apakah semakin lama proses *training*, akan menghasilkan *error* yang semakin baik pula (semakin banyak iterasi, proses *training* menjadi semakin lama).

Pada implementasi analisa ini digunakan model 3 (model optimum) untuk menganalisa penggunaan parameter *epoch* bagi hasil akhir peramalan, yaitu model dengan input (x-1) dan (x-2) dan jumlah *neuron hidden layer* sebanyak 2. Hasil implementasi variasi parameter *epoch* menggunakan model 3, dijelaskan pada Tabel 6.6.

Keterangan:

epoch = jumlah iterasi

$x - n$ = jumlah produksi ikan pada n periode sebelumnya

Tabel 6. 6. Performa Variasi Epoch

Model 3 (x-1 dan y-1) 2 neuron hidden layer								
Epoch	Training MAPE				Testing MAPE			
	Train 1	Train 2	Train 3	average MAPE	Test 1	Test 2	Test 3	average MAPE
1000 dgn val. check 6 (default)	22.6307%	19.8293%	20.8003%	21.0868%	28.0854%	24.4519%	23.1304%	25.2226%
1000	18.9603%	18.9607%	18.0257%	18.6489%	25.4359%	25.4359%	25.6701%	25.5140%
1500	17.6317%	17.6317%	17.6317%	17.6317%	25.8752%	25.8751%	25.8751%	25.9339%
2000	17.7044%	17.2554%	17.4498%	17.4699%	26.2319%	24.6812%	24.6903%	25.2011%
2500	17.4760%	18.0575%	18.0480%	17.8605%	25.5355%	25.7256%	25.4722%	25.5777%
(average max) – (average min)				1.1790%	all test average			25.4899%

Default yang digunakan pada *tool* Matlab adalah sejumlah 1000 iterasi dengan parameter pendukung *validation*

check. *Validation check* merupakan parameter yang digunakan untuk menghentikan iterasi lebih awal apabila *error* yang dihasilkan pada proses iterasi melebihi batas maksimal yang telah ditentukan (batas maksimal *error default* 1.04) sebanyak n kali.

Dilihat pada Tabel 6.6., dari satu sisi disimpulkan bahwa penggunaan variasi *epoch* yang dinilai kurang mempengaruhi hasil akhir peramalan (lihat sub-bab 2.2.5.) yang dilihat dari perbedaan rata-rata *train error* tertinggi dan terendah yang hanya mencapai 1.1790% (hanya parameter *epoch* saja) dan hasil *test error* yang berkisar 25.49%.

Dari sisi lain, seperti yang dijelaskan pada sub-bab sebelumnya, hasil *train error* yang rendah belum tentu menghasilkan hasil *test error* yang rendah pula (*epoch* 2000 menghasilkan *train error* terendah dengan 17.26% dan tidak menghasilkan *test error* terendah pula). Jadi dapat ditarik kesimpulan dari sisi ini, untuk mendapatkan parameter *epoch* yang optimal dapat melalui proses *trial & error*, yaitu dengan mencoba variasi parameter sebanyak mungkin untuk mendapatkan hasil yang bervariasi dan nantinya dibandingkan yang mana merupakan nilai *epoch* optimal.

Umumnya jumlah iterasi yang digunakan pada metode BPNN sebanyak 1000 dengan jumlah *validation check* sebanyak 6 seperti acuan pada Matlab.

6.2.4. Analisa Rasio Dataset

Analisa ini dilakukan untuk melihat pengaruh penggunaan rasio untuk menentukan *train dataset* dan *test dataset* terhadap hasil akhir peramalan. Implementasi untuk analisa menggunakan model 3 (model optimal), yaitu model dengan masukan (x-1) dan (x-2) dengan jumlah *neuron*

hidden layer sebanyak 2. Hasil dari implementasi penggunaan variasi rasio dijelaskan pada Tabel 6.7.

Tabel 6. 7. Performa Variasi Rasio Dataset

Model 3 (x-1 dan y-1) 2 neuron hidden layer							
Rasio dataset (train:test)	Training MAPE			Testing MAPE			
	Train 1	Train 2	Train 3	Test 1	Test 2	Test 3	Average MAPE
85%:15%	20.4001%	20.4002%	20.4002%	30.2930%	30.2930%	30.2930%	30.2930%
80%:20%	28.5249%	28.5249%	19.6443%	28.6357%	28.6357%	27.8751%	28.3822%
75%:25%	24.5641%	24.5641%	22.5883%	30.8724%	30.8724%	32.0431%	31.2626%
70%:30% (default)	22.6307%	19.8293%	20.8003%	28.0854%	24.4519%	23.1304%	25.2226%
60%:40%	23.1838%	22.4821%	22.4821%	47.6254%	37.4693%	37.4693%	40.8546%
50%:50%	16.5616%	15.1525%	15.1525%	33.3118%	25.6820%	25.6820%	28.2252%
40%:60%	15.2766%	15.2766%	16.0947%	31.6264%	31.6263%	30.8929%	31.3819%
(average max) – (average min)							15.6320%

Dilihat pada Tabel 6.7., terdapat perbedaan yang cukup tinggi antara rata-rata *test error* terendah (25.2226%) dan rata-rata *test error* tertinggi (40.8546%) pada *sample* rasio yang digunakan, yaitu sejumlah 15.6320%. Hal ini menyimpulkan bahwa penentuan rasio awal pada *dataset* sewaktu proses perancangan dapat mempengaruhi hasil akhir peramalan (lihat sub-bab 2.2.5).

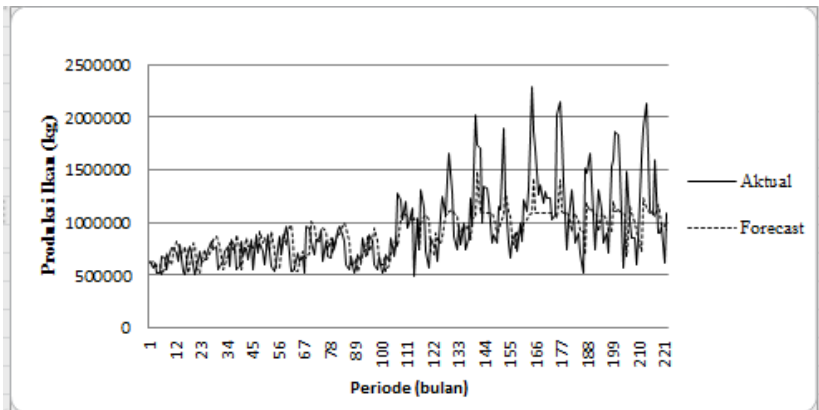
Umumnya digunakan rasio 70% untuk jumlah data *training* dan 30% untuk jumlah data *testing* seperti acuan pada Matlab.

6.3. Hasil Forecast

Hasil implementasi dari peramalan dengan menggunakan model optimal, yaitu model 3 dengan masukan (x-1) dan (x-2) dan jumlah *neuron hidden layer* sebanyak 2 pada seluruh *dataset* yang ada menghasilkan hasil seperti dijelaskan pada Gambar 6.7.

Keterangan:

$x - n$ = jumlah produksi ikan pada n periode sebelumnya



Gambar 6. 7. Hasil *Forecast* menggunakan Model 3

Pada Gambar 6.7. dapat dilihat bahwa nilai hasil peramalan mendekati nilai aktual dari periode 1 hingga periode 124, dan periode 125 hingga seterusnya hasil peramalan dilihat agak menjauhi dari nilai data aktual. Analisa ini dijelaskan pada sub-bab 6.2.1.

Hasil perbedaan yang didapatkan antara nilai data aktual dan nilai hasil ramalan memberikan nilai *error* sebesar 22.49%. Oleh sebab itu, dapat dikatakan metode BPNN dengan menggunakan model 3 layak untuk meramalkan jumlah produksi ikan PPKB (lihat sub-bab 2.2.5.). Didapatkan hasil ramalan untuk jumlah produksi ikan Pelabuhan Perikanan Banjarmasin pada satu periode kedepan (September 2016) adalah 865753 kg.

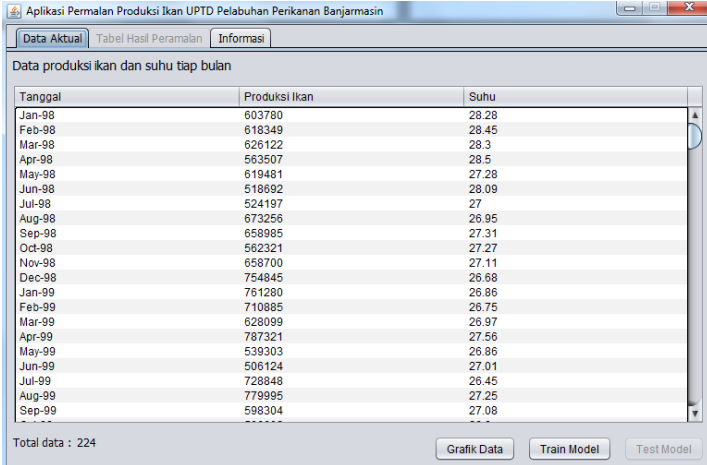
6.4. *Graphical User Interface*

Pada sub-bab ini akan dijelaskan mengenai hasil implementasi fungsional aplikasi yang digambarkan dalam bentuk GUI (*Graphical User Interface*).

6.4.1. Fungsi Menampilkan Data History

Tampilan fungsi untuk menampilkan *data history* bagi pengguna digambarkan pada jendela 'Data Aktual', seperti digambarkan pada Gambar 6.7.

Pada Gambar 6.7. *data history* untuk jumlah produksi ikan Pelabuhan Perikanan Banjarmasin dapat dilihat pada pada kolom 2, sedangkan rata-rata suhu udara stasiun Syamsudin Noor dapat dilihat pada kolom 3, dan bulan/periode waktu kejadian untuk masing-masing variabel dapat dilihat pada kolom 1.



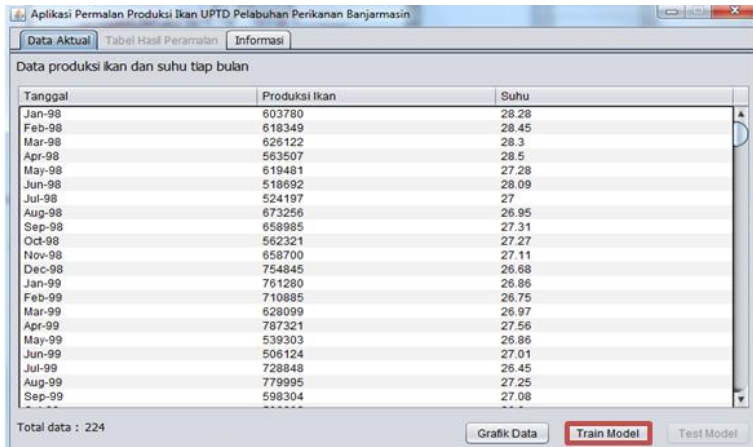
Tanggal	Produksi Ikan	Suhu
Jan-98	603780	28.28
Feb-98	618349	28.45
Mar-98	626122	28.3
Apr-98	563507	28.5
May-98	619481	27.28
Jun-98	518692	28.09
Jul-98	524197	27
Aug-98	673256	26.95
Sep-98	658985	27.31
Oct-98	562321	27.27
Nov-98	658700	27.11
Dec-98	754845	26.68
Jan-99	761280	26.86
Feb-99	710885	26.75
Mar-99	628099	26.97
Apr-99	787321	27.56
May-99	539303	26.86
Jun-99	506124	27.01
Jul-99	728848	26.45
Aug-99	779995	27.25
Sep-99	598304	27.08

Gambar 6. 8. Jendela Tampilan *Data History*

6.4.2. Fungsi Training

Tampilan pengguna untuk melakukan fungsi training digambarkan pada Gambar 6.9. Pengguna dapat melakukan fungsi training menggunakan *dataset* yang ditentukan pada aplikasi dengan menklik tombol 'Train Model' pada jendela 'Data Aktual'. Fungsi training pada aplikasi menggunakan model 3, yaitu model dengan masukan (x-1)

dan (x-2) dengan jumlah *neuron hidden layer* sebanyak 2 dan fungsi aktivasi *tansig* pada *hidden layer* dan fungsi aktivasi *purelin* pada *output layer* dengan menggunakan parameter-parameter yang telah ditentukan (lihat sub-bab 5.1.).



Aplikasi Permalan Produksi Ikan UPTD Pelabuhan Perikanan Banjarmasin

Data Aktual | Tabel Hasil Peramalan | Informasi

Data produksi ikan dan suhu tiap bulan

Tanggal	Produksi Ikan	Suhu
Jan-98	603780	28.28
Feb-98	618349	28.45
Mar-98	626122	28.3
Apr-98	563507	28.5
May-98	619481	27.28
Jun-98	518692	28.09
Jul-98	524197	27
Aug-98	673256	26.95
Sep-98	658985	27.31
Oct-98	562321	27.27
Nov-98	658700	27.11
Dec-98	754845	26.68
Jan-99	761280	26.86
Feb-99	710885	26.75
Mar-99	628099	26.97
Apr-99	787321	27.56
May-99	539303	26.86
Jun-99	506124	27.01
Jul-99	728848	26.45
Aug-99	779995	27.25
Sep-99	598304	27.08

Total data : 224

Grafik Data | **Train Model** | Test Model

Gambar 6. 9. Jendela Fungsi Training

6.4.3. Fungsi Testing

Tampilan pengguna untuk melakukan fungsi testing digambarkan pada Gambar. 6.10. Pengguna dapat melakukan fungsi testing menggunakan *dataset* yang ditentukan dengan menklik tombol 'Test Model' pada jendela 'Data Aktual'. Fungsi testing pada aplikasi ini menggunakan model 3, yaitu model dengan masukan (x-1) dan (x-2) dengan jumlah *neuron hidden layer* sebanyak 2 dan fungsi aktivasi *tansig* pada *hidden layer* dan fungsi aktivasi *purelin* pada *output layer* dan parameter yang telah ditentukan.

Keterangan:

$x - n$ = jumlah produksi ikan pada n periode sebelumnya

6.4.4. Fungsi Menampilkan Testing Error

[illegible]

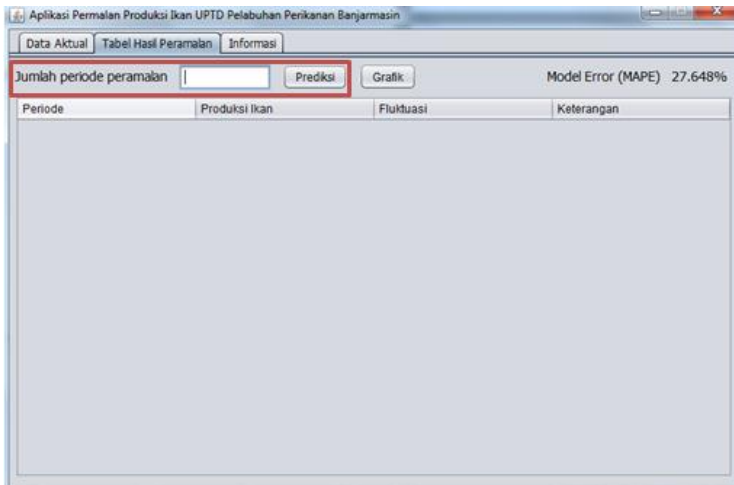
Gambar 6. 11. Jendela Tampilan *Testing Error*

6.4.5. Fungsi Forecast

Tampilan pengguna untuk melakukan fungsi forecast digambarkan pada Gambar 6.11 & Gambar 6.12. Pengguna dapat melakukan fungsi forecast dengan memasukkan jumlah periode peramalan pada kolom 'Jumlah periode peramalan' kemudian mengklik tombol 'Prediksi' pada jendela 'Tabel Hasil Peramalan'. Fungsi forecast pada aplikasi ini menggunakan model 3, yaitu model dengan *input* ($x-1$) dan ($x-2$) dengan jumlah *neuron hidden layer* sebanyak 2 dan fungsi aktivasi *tansig* pada *hidden layer* dan fungsi aktivasi *purelin* pada *output layer* dengan menggunakan parameter yang ditentukan.

Keterangan:

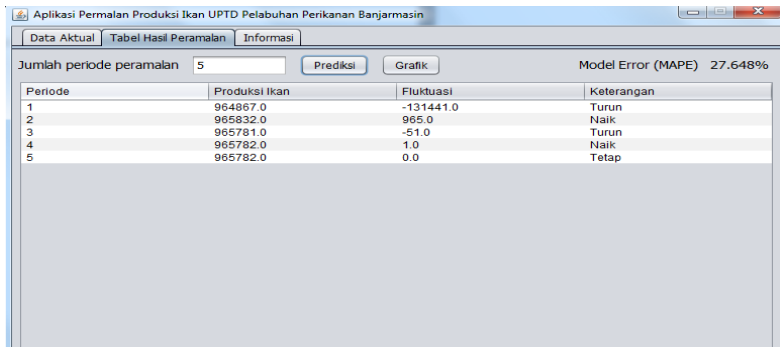
$x - n$ = jumlah produksi ikan pada n periode sebelumnya



Gambar 6. 12. Jendela Fungsi Forecast

Pada Gambar 6.13., terlihat hasil dari proses peramalan menggunakan model 3, yaitu model dengan *input* ($x-1$) dan

(x-2) dengan jumlah *neuron hidden layer* sebanyak 2 yang disajikan dalam bentuk tabel.

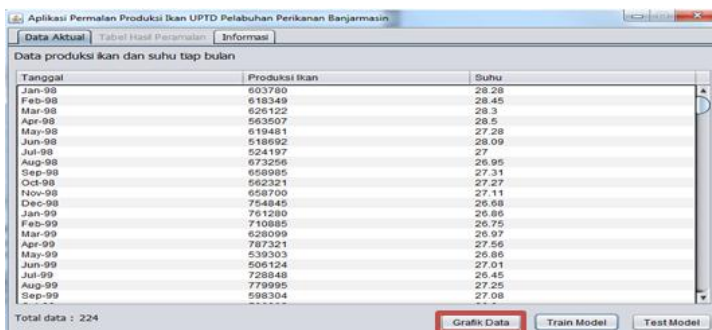


Periode	Produksi Ikan	Fluktuasi	Keterangan
1	964867.0	-131441.0	Turun
2	965832.0	965.0	Naik
3	965781.0	-51.0	Turun
4	965782.0	1.0	Naik
5	965782.0	0.0	Tetap

Gambar 6. 13. Jendela Tabel Hasil *Forecast*

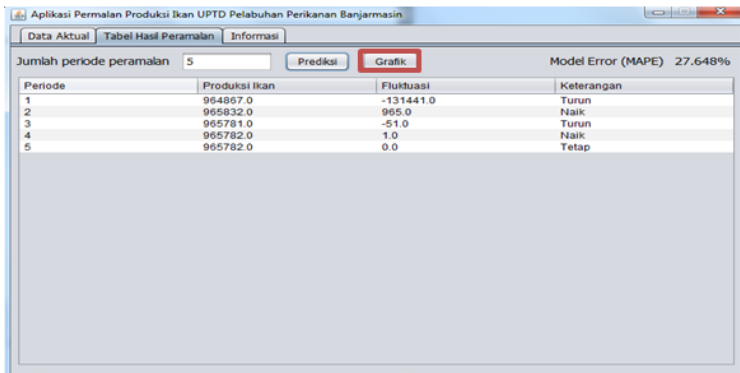
6.4.6. Fungsi Menampilkan Grafik

Tampilan pengguna untuk melihat tampilan grafik *data history* dan data hasil peramalan dengan mengklik tombol 'Grafik Data' pada jendela 'Data Aktual' dan tombol 'Grafik' pada jendela 'Tabel Hasil Peramalan', seperti pada Gambar 6.14. dan Gambar 6.15.



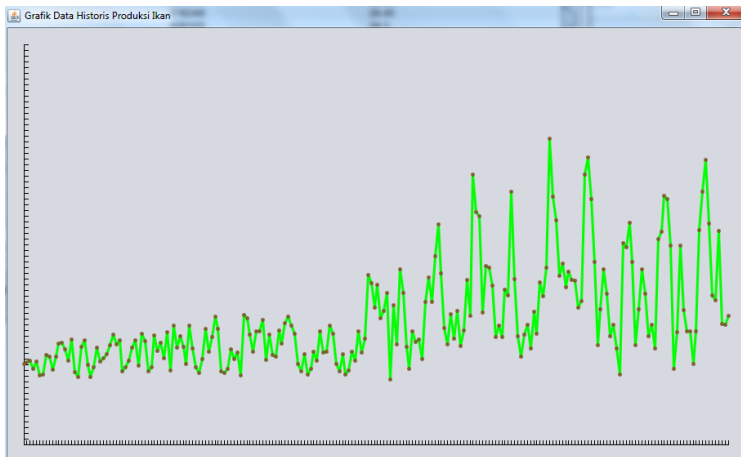
Tanggal	Produksi Ikan	Suhu
Jan-98	603760	28.28
Feb-98	618349	28.45
Mar-98	626122	28.3
Apr-98	663507	28.5
May-98	619481	27.28
Jun-98	518692	28.09
Jul-98	524197	27
Aug-98	673256	26.95
Sep-98	658985	27.31
Oct-98	662321	27.27
Nov-98	658700	27.11
Dec-98	754845	26.68
Jan-99	761280	26.86
Feb-99	710885	26.75
Mar-99	628099	26.97
Apr-99	787321	27.56
May-99	539303	26.80
Jun-99	506124	27.01
Jul-99	728848	26.45
Aug-99	779995	27.25
Sep-99	598304	27.08

Gambar 6. 14. Fungsi Grafik *Data History*



Gambar 6. 15. Fungsi Grafik Hasil *Forecast*

Gambar 6.16. merupakan pengaplikasian fungsi Menampilkan Grafik oleh pengguna, yaitu sebelumnya menklik tombol 'Grafik Data' pada jendela 'Data Aktual' (lihat gambar 6.14.).



Gambar 6. 16. Grafik *Data History*

“Halaman ini sengaja dikosongkan”

BAB VII

KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan dan saran dari sudut pandang yang diambil dari hasil implementasi dan analisa mengenai penggunaan metode BPNN pada *dataset* jumlah produksi ikan UPTD Pelabuhan Perikanan Banjarmasin pada periode Januari 1998 – Agustus 2016.

7.1. Kesimpulan

Dari hasil proses analisa dan diskusi, berdasarkan hasil penelitian ini maka dapat disimpulkan bahwa:

1. Metode BPNN dengan masukan satu dan dua periode sebelum sekarang layak digunakan terhadap *dataset* seperti *dataset* jumlah produksi ikan pada UPTD Pelabuhan Perikanan Banjarmasin dengan melihat pada hasil *error* yang dihasilkan berkisar 20% pada proses *testing* maupun peramalan pada seluruh *data history*.
2. Variabel pendukung, yaitu suhu udara dinilai kurang mempengaruhi terhadap hasil akhir peramalan, yaitu dilihat dari dekatnya hasil MAPE yang dihasilkan (< 5%) oleh model yang menggunakan variabel pendukung maupun model yang tidak menggunakan variabel pendukung.
3. Variasi Parameter *learning rate* dan *epoch* dinilai kurang mempengaruhi terhadap hasil akhir peramalan, sedangkan penggunaan variasi rasio data *train & test* dinilai mempengaruhi hasil akhir peramalan.
4. Kekurangan yang seperti dijelaskan pada poin 1 dan poin 2 mungkin dikarenakan adanya perbedaan pola data antara *data training* dan *data testing*. Pada pola data training dapat dilihat umumnya berpola *horizontal* dan pada data testing berpola *seasonal*.

7.2. Saran

Untuk penelitian selanjutnya dengan topik terkait, berdasarkan hasil dan analisa pada penelitian ini disarankan untuk:

1. Gunakan *dataset* yang tidak memiliki nilai fluktuasi terlalu tinggi atau berpola acak atau memiliki pola lebih dari satu, karena mungkin dapat mempengaruhi hasil akhir peramalan.
2. Gunakan nilai *default*, yaitu nilai parameter *learning rate* 0.01 dan *epoch* 1000 dengan *val. check* 6 dan untuk penentuan rasio *dataset* gunakan rasio 70:30 untuk data *train* dan *test* apabila menggunakan tool Matlab.
3. Untuk mendapatkan model optimal dapat juga menggunakan metode *trial & error*, yaitu menggunakan variasi parameter sebanyak mungkin dan kemudian dibandingkan hasil performanya.
4. Penelitian selanjutnya dapat menggunakan variabel pendukung lainnya, seperti kadar garam pada air laut, kecepatan arus, ataupun massa air laut (keadaan pasang/surut).
5. Metode BPNN dapat dikombinasikan dengan metode peramalan lainnya dan nantinya dibandingkan untuk melihat hasil kinerjanya.

DAFTAR PUSTAKA

- [1] Barry Render and Jay Heizer, *Prinsip-prinsip Manajemen Operasi*, PT. Salemba Emban Patria, Jakarta, 2001.
- [2] Szkuta, B. R., Sanabria, L. A., & Dillon, T. S. (1999). Electricity price short-term forecasting using artificial neural networks. *IEEE transactions on power systems*, 14(3), 851-857.
- [3] Mansur, A., & Kuncoro, T. (2012). Product inventory predictions at small medium enterprise using market basket analysis approach-neural networks. *Procedia Economics and Finance*, 4, 312-320.
- [4] Ihwan, A., 2013. Metode Jaringan Saraf Tiruan Propagasi Balik Untuk Estimasi Curah Hujan Bulanan di Ketapang Kalimantan Barat. *Prosiding SEMIRATA 2013*, 1(1).
- [5] Produksi, M. Analisis dan Implementasi Jaringan Syaraf Tiruan–Propagasi Balik Dalam Memprediksi Produksi dan Konsumsi Minyak Bumi, Gas Bumi, dan Batu Bara di Indonesia Analysis and Implementation of Artificial Neural Network–Back Propagation in Prediction of Crude Oil, Natural Gas, and Coal Production and Consumption in
- [6] Heravi, S., Osborn, D. R., & Birchenhall, C. R. (2004). Linear versus neural network forecasts for European industrial production series. *International Journal of Forecasting*, 20(3), 435-446.
- [7] Sridhar, V. N., Dadhwal, V. K., Chaudhari, K. N., Sharma, R., Bairagi, G. D., & Sharma, A. K. (1994). Wheat production forecasting for a predominantly unirrigated region in Madhya Pradesh (India). *TitleREMOTE SENSING*, 15(6), 1307-1316.
- [8] “Dinas Perikanan dan Kelautan.” [Online]. Available: <http://diskanlut.kalselprov.go.id/>. [Accessed: 25-May-2016].

- [9] “Profil Pelabuhan Perikanan Pantai Banjarmasin,” *Alam Ikan*. [Online]. Available: http://www.alamikan.com/2012/11/mengetahui-profil-pelabuhan-perikanan_99.html. [Accessed: 25-May-2016].
- [10] Hela, I. and Laevastu, T., 1970. Fisheries oceanography. *Fishing News (Books) LTD, London*.
- [11] Nybakken, J.W. and Nybakken, J.W., 1993. *Marine biology: an ecological approach* (No. QH 91. N93 1993).
- [12] McCulloch, W.S. and Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), pp.115-133.
- [13] DARPA Neural Network Study, AFCEA International Press, 1998
- [14] rezahaikal, “JARINGAN SARAF TIRUAN (ARTIFICIAL NEURAL NETWORK),” *rezkal*, 17-Jan-2013.
- [15] Makridakis, S., Wheelwright, S.C. and Hyndman, R.J., 2008. *Forecasting methods and applications*. John Wiley & Sons.
- [16] Pramana, I.P.A.A., 2016. Peramalan Jumlah Kasus Demam Berdarah di Kabupaten Malang Menggunakan Metode Fuzzy Inference System. *Jurnal Teknik ITS*, 5(1).
- [17] “National Centers for Environmental Information | National Centers for Environmental Information (NCEI) formerly known as National Climatic Data Center (NCDC).” [Online]. Available: <https://www.ncdc.noaa.gov/news/national-centers-environmental-information>. [Accessed: 05-Jan-2017].
- [18] “Multilayer Neural Network Architecture - MATLAB & Simulink.” [Online]. Available: <https://www.mathworks.com/help/nnet/ug/multilayer->

- neural-network-architecture.html. [Accessed: 05-Jan-2017].
- [19] tutorialspoint.com, “Java Tutorial,” *www.tutorialspoint.com*. [Online]. Available: <http://www.tutorialspoint.com/java/>. [Accessed: 12-Jan-2017].
- [20] “JavaScript and HTML DOM Reference.” [Online]. Available: <http://www.w3schools.com/jsref/>. [Accessed: 12-Jan-2017].
- [21] “Overview (Java Platform SE 7).” [Online]. Available: <http://docs.oracle.com/javase/7/docs/api/>. [Accessed: 12-Jan-2017].

“Halaman ini sengaja dikosongkan”

BIODATA PENULIS



Penulis dilahirkan pada tanggal 4 Agustus 1994 di Kota Banjarbaru, Kalimantan Selatan. Penulis merupakan anak ke-4 dari empat bersaudara. Penulis telah menempuh pendidikan formal pada SDN Utara 4 Banjarbaru, SMPN 1 Banjarbaru, dan SMAN 1 Banjarbaru. Setelah itu, penulis melanjutkan pendidikan jenjang S-1 pada Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember.

Selama menempuh pendidikan pada jenjang S-1, penulis cukup aktif dalam berorganisasi dan mengikuti kegiatan ekstrakurikuler di lingkungan kampus ITS. Pada semester sembilan, penulis mengambil topik teknik peramalan yang diangkat sebagai judul tugas akhir pada Lab Rekayasa Data dan Inteligensi Bisnis dibawah bimbingan Bapak Edwin Riksakomara, S. Kom., M. T. Selama satu semester pengerjaan, penulis dapat menyelesaikan Tugas Akhir dengan judul “Peramalan Jumlah Produksi Ikan dengan Menggunakan *Backpropagation Neural Network* (Studi Kasus: UPTD Pelabuhan Perikanan Banjarmasin)”. Semoga hasil dari pengerjaan Tugas Akhir ini dapat bermanfaat bagi semua pihak. Amin.

Untuk mendapatkan informasi lebih lanjut mengenai penulis dapat menghubungi *e-mail* asar.razak@gmail.com.

“Halaman ini sengaja dikosongkan”

LAMPIRAN A
DATA INPUT

Periode	Produksi Ikan (kg)	Suhu Udara (°C)
Jan-98	603780	28.28
Feb-98	618349	28.45
Mar-98	626122	28.30
Apr-98	563507	28.50
May-98	619481	27.28
Jun-98	518692	28.09
Jul-98	524197	27.00
Aug-98	673256	26.95
Sep-98	658985	27.31
Oct-98	562321	27.27
Nov-98	658700	27.11
Dec-98	754845	26.68
Jan-99	761280	26.86
Feb-99	710885	26.75
Mar-99	628099	26.97
Apr-99	787321	27.56
May-99	539303	26.86
Jun-99	506124	27.01
Jul-99	728848	26.45
Aug-99	779995	27.25
Sep-99	598304	27.08

Periode	Produksi Ikan (kg)	Suhu Udara (°C)
Oct-99	506602	26.80
Nov-99	577256	26.94
Dec-99	728099	26.87
Jan-00	622894	26.65
Feb-00	644349	27.03
Mar-00	679231	27.26
Apr-00	744117	27.57
May-00	826489	27.74
Jun-00	749638	26.93
Jul-00	780467	27.20
Aug-00	550825	27.16
Sep-00	579929	28.03
Oct-00	629782	27.68
Nov-00	725640	27.55
Dec-00	781999	27.53
Jan-01	593481	27.02
Feb-01	832070	27.53
Mar-01	774545	27.45
Apr-01	548488	28.08
May-01	579502	28.52
Jun-01	815838	27.87
Jul-01	699191	28.05

Periode	Produksi Ikan (kg)	Suhu Udara (°C)
Aug-01	762882	28.47
Sep-01	643234	28.57
Oct-01	839434	28.01
Nov-01	556999	27.37
Dec-01	892240	27.56
Jan-02	723190	27.34
Feb-02	811043	27.51
Mar-02	729906	27.47
Apr-02	601141	28.01
May-02	889076	28.33
Jun-02	720544	27.39
Jul-02	580062	27.97
Aug-02	532998	27.83
Sep-02	640955	28.57
Oct-02	864925	29.22
Nov-02	696067	27.40
Dec-02	806912	27.57
Jan-03	956791	27.12
Feb-03	868772	27.14
Mar-03	545514	27.29
Apr-03	534699	27.53
May-03	567055	27.73

Periode	Produksi Ikan (kg)	Suhu Udara (°C)
Jun-03	711573	27.39
Jul-03	640512	26.86
Aug-03	688291	27.78
Sep-03	516427	27.80
Oct-03	969267	27.46
Nov-03	945138	27.15
Dec-03	821993	26.43
Jan-04	693134	26.85
Feb-04	846338	27.05
Mar-04	850345	27.37
Apr-04	936688	27.75
May-04	634326	28.03
Jun-04	825178	27.58
Jul-04	672960	26.74
Aug-04	658795	27.19
Sep-04	853490	28.36
Oct-04	758259	28.48
Nov-04	910175	27.08
Dec-04	960472	26.78
Jan-05	892032	26.86
Feb-05	829220	27.17
Mar-05	602430	27.20

Periode	Produksi Ikan (kg)	Suhu Udara (°C)
Apr-05	547800	27.32
May-05	677828	27.28
Jun-05	526170	27.54
Jul-05	564659	27.03
Aug-05	692035	27.52
Sep-05	628802	28.33
Oct-05	849643	26.85
Nov-05	685934	27.01
Dec-05	695140	26.58
Jan-06	892032	26.45
Feb-06	829220	26.87
Mar-06	602430	27.42
Apr-06	547800	27.56
May-06	677828	27.68
Jun-06	526170	26.37
Jul-06	553380	27.10
Aug-06	692035	27.04
Sep-06	628802	27.53
Oct-06	849643	28.45
Nov-06	685934	27.99
Dec-06	790930	27.85
Jan-07	1272948	27.18

Periode	Produksi Ikan (kg)	Suhu Udara (°C)
Feb-07	1209825	26.59
Mar-07	1026675	27.20
Apr-07	1199120	27.29
May-07	943155	27.68
Jun-07	1003082	27.00
Jul-07	1137474	26.71
Aug-07	488293	26.65
Sep-07	1041426	27.45
Oct-07	747409	27.65
Nov-07	1315274	26.67
Dec-07	1137474	27.03
Jan-08	728746	27.40
Feb-08	564245	27.46
Mar-08	850441	27.02
Apr-08	768789	27.31
May-08	785808	27.94
Jun-08	637423	27.51
Jul-08	1066476	26.06
Aug-08	1250475	26.43
Sep-08	1069219	27.59
Oct-08	1409791	26.96
Nov-08	1651882	27.27

Periode	Produksi Ikan (kg)	Suhu Udara (°C)
Dec-08	1280149	26.22
Jan-09	873906	26.45
Feb-09	748426	26.82
Mar-09	973508	27.44
Apr-09	791560	28.06
May-09	1001392	27.46
Jun-09	736367	27.86
Jul-09	854281	26.95
Aug-09	1234814	27.76
Sep-09	964040	28.79
Oct-09	2024032	28.00
Nov-09	1741315	27.89
Dec-09	1713661	27.47
Jan-10	988692	27.08
Feb-10	1335362	28.16
Mar-10	1325262	27.72
Apr-10	1193555	28.36
May-10	805305	28.56
Jun-10	889795	27.25
Jul-10	807118	27.01
Aug-10	1159590	27.31
Sep-10	1119370	27.22

Periode	Produksi Ikan (kg)	Suhu Udara (°C)
Oct-10	1894362	27.52
Nov-10	1239350	27.51
Dec-10	812800	26.89
Jan-11	657500	26.96
Feb-11	820700	27.39
Mar-11	896000	27.10
Apr-11	721600	27.34
May-11	997000	28.36
Jun-11	830000	27.93
Jul-11	1216600	27.62
Aug-11	1108800	28.42
Sep-11	1327428	27.84
Oct-11	2295195	28.01
Nov-11	1858050	27.70
Dec-11	1683468	26.31
Jan-12	1261425	27.01
Feb-12	1358602	27.16
Mar-12	1180180	27.37
Apr-12	1296550	27.51
May-12	1235296	27.69
Jun-12	1228282	26.99
Jul-12	1023025	26.12

Periode	Produksi Ikan (kg)	Suhu Udara (°C)
Aug-12	1076250	27.24
Sep-12	2021140	28.17
Oct-12	2154850	28.24
Nov-12	1839396	27.78
Dec-12	1367300	27.12
Jan-13	746200	27.10
Feb-13	1010800	27.30
Mar-13	1314100	27.68
Apr-13	1127000	28.13
May-13	812500	27.97
Jun-13	896130	28.36
Jul-13	719614	26.66
Aug-13	520750	27.11
Sep-13	1510812	27.68
Oct-13	1476478	28.27
Nov-13	1662250	27.52
Dec-13	1367300	26.77
Jan-14	746200	27.08
Feb-14	1010800	27.20
Mar-14	1314100	27.27
Apr-14	1127000	27.70
May-14	812500	28.08

Periode	Produksi Ikan (kg)	Suhu Udara (°C)
Jun-14	896130	27.63
Jul-14	719614	27.76
Aug-14	1539750	27.51
Sep-14	1596000	28.58
Oct-14	1863667	29.56
Nov-14	1839600	28.11
Dec-14	1493784	27.11
Jan-15	566000	26.69
Feb-15	839900	26.78
Mar-15	1492450	27.44
Apr-15	1007865	28.03
May-15	847721	28.33
Jun-15	850465	27.95
Jul-15	603695	28.30
Aug-15	849716	28.48
Sep-15	1607265	28.98
Oct-15	1863667	29.93
Nov-15	1839600	28.65
Dec-15	1493784	27.79
Jan-16	566000	28.25
Feb-16	839900	27.77
Mar-16	1492450	28.19

Periode	Produksi Ikan (kg)	Suhu Udara (°C)
Apr-16	1007865	28.34
May-16	847721	28.78
Jun-16	850465	27.70
Jul-16	603695	27.82
Aug-16	849716	28.28

“Halaman ini sengaja dikosongkan”

LAMPIRAN B
TRAINING PROCESS OUTPUT OPTIMUM MODEL

Periode	Target
3	626122
4	563507
5	619481
6	518692
7	524197
8	673256
9	658985
10	562321
11	658700
12	754845

Train 1		Train 2		Train 3	
Error	APE	Error	APE	Error	APE
-177510	28.35%	-131710	21.04%	-148761	23.76%
-239777	42.55%	-195076	34.62%	-212425	37.70%
-175859	28.39%	-124639	20.12%	-139534	22.52%
-289072	55.73%	-242125	46.68%	-259133	49.96%
-266962	50.93%	-209836	40.03%	-222597	42.46%
-126161	18.74%	-68636.9	10.19%	-81748.2	12.14%
-162559	24.67%	-117286	17.80%	-135728	20.60%
-240138	42.70%	-199794	35.53%	-218346	38.83%
-134179	20.37%	-83084.2	12.61%	-97915.4	14.86%
-59060.3	7.82%	-15009.4	1.99%	-33235.7	4.40%

B-2

Periode	Target
13	761280
14	710885
15	628099
16	787321
17	539303
18	506124
19	728848
20	779995
21	598304
22	506602
23	577256
24	728099

Train 1		Train 2		Train 3	
Error	APE	Error	APE	Error	APE
-56696.2	7.45%	-23776.5	3.12%	-44902.1	5.90%
-97096.3	13.66%	-67910.2	9.55%	-89611.3	12.61%
-173034	27.55%	-139249	22.17%	-159578	25.41%
-8657.63	1.10%	34254.2	4.35%	16725.81	2.12%
-289434	53.67%	-257107	47.67%	-278732	51.68%
-277050	54.74%	-222321	43.93%	-235890	46.61%
-66757.7	9.16%	-7331.35	1.01%	-19534.2	2.68%
-56756.3	7.28%	-12297	1.58%	-31816.9	4.08%
-215180	35.97%	-186947	31.25%	-209012	34.93%
-282074	55.68%	-235842	46.55%	-252184	49.78%
-214110	37.09%	-155192	26.88%	-167349	28.99%
-79561.9	10.93%	-26995.7	3.71%	-42367.2	5.82%

25	622894
26	644349
27	679231
28	744117
29	826489
30	749638
31	780467
32	550825
33	579929
34	629782
35	725640
36	781999
37	593481

-201557	32.36%	-162540	26.09%	-182583	29.31%
-149889	23.26%	-106540	16.53%	-123881	19.23%
-125997	18.55%	-83070.4	12.23%	-101042	14.88%
-63734.6	8.57%	-24426.9	3.28%	-43531.2	5.85%
12831.75	1.55%	45734.5	5.53%	24764.15	3.00%
-69622.2	9.29%	-46026.2	6.14%	-69254.1	9.24%
-19788.5	2.54%	9352.368	1.20%	-12222.4	1.57%
-260399	47.27%	-232818	42.27%	-254984	46.29%
-204592	35.28%	-151698	26.16%	-165860	28.60%
-173808	27.60%	-122941	19.52%	-138483	21.99%
-81882.4	11.28%	-36363.1	5.01%	-53755	7.41%
-34682.6	4.44%	1642.477	0.21%	-18642.5	2.38%
-220691	37.19%	-192527	32.44%	-214626	36.16%

B-4

Periode	Target
38	832070
39	774545
40	548488
41	579502
42	815838
43	699191
44	762882
45	643234
46	839434
47	556999
48	892240
49	723190

Train 1		Train 2		Train 3	
Error	APE	Error	APE	Error	APE
43935.32	5.28%	90805.97	10.91%	74671.19	8.97%
-71859.4	9.28%	-39477.1	5.10%	-61577.8	7.95%
-254111	46.33%	-227524	41.48%	-249797	45.54%
-205136	35.40%	-151935	26.22%	-165990	28.64%
12082.09	1.48%	63056.63	7.73%	47534.22	5.83%
-145671	20.83%	-111185	15.90%	-132874	19.00%
-32925.4	4.32%	1478.744	0.19%	-18573.9	2.43%
-171038	26.59%	-140367	21.82%	-161884	25.17%
45536.48	5.42%	86353.77	10.29%	68241.08	8.13%
-281156	50.48%	-253310	45.48%	-276052	49.56%
110250.3	12.36%	162752.4	18.24%	148373.8	16.63%
-153306	21.20%	-119636	16.54%	-142047	19.64%

50	811043
51	729906
52	601141
53	889076
54	720544
55	580062
56	532998
57	640955
58	864925
59	696067
60	806912
61	956791
62	868772

17915.26	2.21%	49484	6.10%	28610.11	3.53%
-89413	12.25%	-63663	8.72%	-86421.8	11.84%
-198169	32.97%	-166890	27.76%	-187870	31.25%
97084.34	10.92%	142998.1	16.08%	126517.7	14.23%
-141949	19.70%	-113600	15.77%	-136666	18.97%
-213021	36.72%	-181163	31.23%	-201952	34.82%
-257586	48.33%	-208994	39.21%	-224583	42.14%
-154517	24.11%	-99027.5	15.45%	-112536	17.56%
50351.6	5.82%	97010.22	11.22%	79414.76	9.18%
-149297	21.45%	-123095	17.68%	-146271	21.01%
14555.27	1.80%	49124.28	6.09%	29134.46	3.61%
134492.9	14.06%	161728.1	16.90%	139223.4	14.55%
32638.52	3.76%	39278.61	4.52%	12469.18	1.44%

B-6

Periode	Target
63	545514
64	534699
65	567055
66	711573
67	640512
68	688291
69	516427
70	969267
71	945138
72	821993
73	693134
74	846338

Train 1		Train 2		Train 3	
Error	APE	Error	APE	Error	APE
-257523	47.21%	-241852	44.33%	-266928	48.93%
-244900	45.80%	-190213	35.57%	-203961	38.15%
-231357	40.80%	-175568	30.96%	-189169	33.36%
-91847.1	12.91%	-39278.1	5.52%	-54287	7.63%
-182196	28.45%	-141636	22.11%	-161244	25.17%
-108949	15.83%	-67385.7	9.79%	-85365.1	12.40%
-293120	56.76%	-254425	49.27%	-273777	53.01%
182631.1	18.84%	240184	24.78%	227646.1	23.49%
14418.77	1.53%	52081.67	5.51%	29853.64	3.16%
10616.88	1.29%	16531.38	2.01%	-10765.7	1.31%
-105846	15.27%	-84919.6	12.25%	-108698	15.68%
51535.42	6.09%	86550.55	10.23%	66686.14	7.88%

75	850345
76	936688
77	634326
78	825178
79	672960
80	658795
81	853490
82	758259
83	910175
84	960472
85	892032
86	829220
87	602430

19553.01	2.30%	43693.3	5.14%	20379.8	2.40%
125748.6	13.42%	144358.7	15.41%	120077.5	12.82%
-190378	30.01%	-182188	28.72%	-208746	32.91%
41735.32	5.06%	84081	10.19%	66356.47	8.04%
-163358	24.27%	-133899	19.90%	-156297	23.23%
-133904	20.33%	-96686.9	14.68%	-115880	17.59%
51028.24	5.98%	91396.65	10.71%	72851.21	8.54%
-80428.8	10.61%	-54720.3	7.22%	-77872.1	10.27%
111052.7	12.20%	139082.7	15.28%	117215.3	12.88%
126516.3	13.17%	141086.7	14.69%	115854.1	12.06%
71312.83	7.99%	75188.83	8.43%	47628.15	5.34%
23818.43	2.87%	36713.94	4.43%	10986.04	1.32%
-201468	33.44%	-181110	30.06%	-204981	34.03%

B-8

Periode	Target
88	547800
89	677828
90	526170
91	564659
92	692035
93	628802
94	849643
95	685934
96	695140
97	892032
98	829220
99	602430

Train 1		Train 2		Train 3	
Error	APE	Error	APE	Error	APE
-238495	43.54%	-192633	35.16%	-209118	38.17%
-117578	17.35%	-64185.3	9.47%	-78378.4	11.56%
-292911	55.67%	-249311	47.38%	-267993	50.93%
-223503	39.58%	-167467	29.66%	-180533	31.97%
-111882	16.17%	-58832.9	8.50%	-73735.2	10.65%
-190563	30.31%	-148748	23.66%	-167875	26.70%
52193.1	6.14%	95244.93	11.21%	77704.85	9.15%
-157794	23.00%	-129534	18.88%	-152313	22.21%
-97233.7	13.99%	-61517.6	8.85%	-81163.2	11.68%
86146.43	9.66%	122909.1	13.78%	103221.2	11.57%
-12198.9	1.47%	7759.603	0.94%	-16499.2	1.99%
-201468	33.44%	-181110	30.06%	-204981	34.03%

100	547800
101	677828
102	526170
103	553380
104	692035
105	628802
106	849643
107	685934
108	790930
109	1272948
110	1209825
111	1026675
112	1199120

-238495	43.54%	-192633	35.16%	-209118	38.17%
-117578	17.35%	-64185.3	9.47%	-78378.4	11.56%
-292911	55.67%	-249311	47.38%	-267993	50.93%
-234782	42.43%	-178746	32.30%	-191812	34.66%
-110394	15.95%	-56224.1	8.12%	-70655	10.21%
-192137	30.56%	-149667	23.80%	-168735	26.83%
52193.1	6.14%	95244.93	11.21%	77704.85	9.15%
-157794	23.00%	-129534	18.88%	-152313	22.21%
-1443.69	0.18%	34272.35	4.33%	14626.8	1.85%
452194.3	35.52%	481159.5	37.80%	459066.7	36.06%
215818.3	17.84%	138302.3	11.43%	99280.52	8.21%
210209.1	20.47%	167378.7	16.30%	127297.4	12.40%
398704.2	33.25%	395918.8	33.02%	365701.1	30.50%

Periode	Target
113	943155
114	1003082
115	1137474
116	488293
117	1041426
118	747409
119	1315274
120	1137474
121	728746
122	564245
123	850441
124	768789

Train 1		Train 2		Train 3	
Error	APE	Error	APE	Error	APE
87187.85	9.24%	32779.28	3.48%	-6625.64	0.70%
209130.9	20.85%	218039	21.74%	190685.7	19.01%
313589.6	27.57%	310344.9	27.28%	281240.8	24.73%
-355219	72.75%	-389297	79.73%	-424711	86.98%
280264	26.91%	350907.8	33.69%	340958.7	32.74%
-250789	33.55%	-208988	27.96%	-230674	30.86%
528197.6	40.16%	557633.8	42.40%	535966.8	40.75%
73179.1	6.43%	-12935.6	1.14%	-51517	4.53%
-74947.1	10.28%	-94970.9	13.03%	-130027	17.84%
-217228	38.50%	-184798	32.75%	-205830	36.48%
61822.61	7.27%	112527.6	13.23%	97658.88	11.48%
-90261.2	11.74%	-55734	7.25%	-77739.4	10.11%

125	785808
126	637423
127	1066476
128	1250475
129	1069219
130	1409791
131	1651882
132	1280149
133	873906
134	748426
135	973508
136	791560
137	1001392

-14669.2	1.87%	12311.46	1.57%	-9844.29	1.25%
-172528	27.07%	-145922	22.89%	-168296	26.40%
274606.2	25.75%	315999.5	29.63%	298095	27.95%
318042.7	25.43%	322848.8	25.82%	295955.9	23.67%
206594.4	19.32%	132541.6	12.40%	89057.79	8.33%
608137.8	43.14%	599269.5	42.51%	567378.6	40.25%
715705	43.33%	549920.4	33.29%	493083.7	29.85%
358259.9	27.99%	29639.62	2.32%	-55904.6	4.37%
78465.58	8.98%	46518.38	5.32%	4999.792	0.57%
-37001.9	4.94%	-18884.2	2.52%	-44127.5	5.90%
176837.3	18.16%	205764.8	21.14%	184148.9	18.92%
-61300.9	7.74%	-54210.7	6.85%	-80887.8	10.22%
207130.2	20.68%	231374.6	23.11%	208413.3	20.81%

B-12

Periode	Target
138	736367
139	854281
140	1234814
141	964040
142	2024032
143	1741315
144	1713661
145	988692
146	1335362
147	1325262
148	1193555
149	805305

Train 1		Train 2		Train 3	
Error	APE	Error	APE	Error	APE
-114842	15.60%	-114405	15.54%	-142338	19.33%
66084.85	7.74%	96501.78	11.30%	75178.6	8.80%
409263.8	33.14%	430581	34.87%	406777.3	32.94%
32189.76	3.34%	-35911.1	3.73%	-75018.8	7.78%
1230364	60.79%	1237285	61.13%	1209316	59.75%
-46833	2.69%	-112274	6.45%	-118877	6.83%
902649.3	52.67%	678717.3	39.61%	572712.7	33.42%
144311.5	14.60%	-155151	15.69%	-258143	26.11%
558065.5	41.79%	570908.6	42.75%	543340.5	40.69%
396077.1	29.89%	274835.2	20.74%	225787.7	17.04%
365601.9	30.63%	281738.7	23.61%	231460.3	19.39%
-3717.22	0.46%	-37588.9	4.67%	-76136.3	9.45%

150	889795		105371.6	11.84%	129795.1	14.59%	106436.7	11.96%
151	807118		-15201.9	1.88%	-43.6788	0.01%	-25116.2	3.11%
152	1159590		358011.5	30.87%	380733.2	32.83%	357455.8	30.83%
153	1119370		209262.2	18.69%	171628.5	15.33%	137476.8	12.28%
154	1894362		1078546	56.93%	1055083	55.70%	1020472	53.87%
155	1239350		-211789	17.09%	-498275	40.20%	-528295	42.63%
156	812800		27835.98	3.42%	18986.89	2.34%	-16520	2.03%
		average:		22.6307%		19.8293%		20.8003%

“Halaman ini sengaja dikosongkan”

LAMPIRAN C
TESTING PROCESS OUTPUT OPTIMUM MODEL

Periode	Target
159	896000
160	721600
161	997000
162	830000
163	1216600
164	1108800
165	1327428
166	2295195
167	1858050
168	1683468

Test 1		Test 2		Test 3	
Error	APE	Error	APE	Error	APE
64861.05	7.24%	93089.23	10.39%	70576.27	7.88%
-99722.2	13.82%	-85698.1	11.88%	-111014	15.38%
204241.8	20.49%	235981.6	23.67%	215154.9	21.58%
-37681.1	4.54%	-32096.2	3.87%	-59021.2	7.11%
420354.3	34.55%	440468.7	36.20%	416407.5	34.23%
176495.7	15.92%	117299.9	10.58%	79795.73	7.20%
518805.6	39.08%	500425.5	37.70%	466516.2	35.14%
1416276	61.71%	1305458	56.88%	1254945	54.68%
84403.39	4.54%	-30759.8	1.66%	-33208.3	1.79%
880787.8	52.32%	640970.7	38.07%	517193.7	30.72%

C-2

Periode	Target
169	1261425
170	1358602
171	1180180
172	1296550
173	1235296
174	1228282
175	1023025
176	1076250
177	2021140
178	2154850
179	1839396
180	1367300

Test 1		Test 2		Test 3	
Error	APE	Error	APE	Error	APE
441206	34.98%	215796	17.11%	117657.3	9.33%
565692.2	41.64%	540834.5	39.81%	501244	36.89%
330966.7	28.04%	218809.2	18.54%	164428.3	13.93%
491771.1	37.93%	463891.3	35.78%	426497.4	32.89%
385797.7	31.23%	298406.4	24.16%	250452.6	20.27%
410686.8	33.44%	361019.6	29.39%	319068	25.98%
199815.5	19.43%	146481	14.32%	104750.2	10.24%
277381.4	25.77%	275707.1	25.62%	245666	22.83%
1194950	59.12%	1177518	58.26%	1145169	56.66%
473510.8	21.97%	321939.2	14.94%	309734	14.37%
902937.6	49.09%	212174	11.53%	126446.9	6.87%
556539.9	40.70%	279028.7	20.41%	155578.6	11.38%

181	746200
182	1010800
183	1314100
184	1127000
185	812500
186	896130
187	719614
188	520750
189	1510812
190	1476478
191	1662250
192	1367300
193	746200

-46621.6	6.25%	-85646.2	11.48%	-131755	17.66%
237268.8	23.47%	269997.2	26.71%	248541.1	24.59%
448327.8	34.12%	449891.6	34.24%	422247.6	32.13%
219195.2	19.45%	110331.6	9.79%	62403.25	5.54%
9759.959	1.20%	-7978.23	0.98%	-42459.5	5.23%
108300.4	12.09%	131258.7	14.65%	107663.2	12.01%
-102868	14.29%	-88640.7	12.32%	-113913	15.83%
-271824	52.20%	-239868	46.06%	-260632	50.05%
725582.7	48.03%	782654.8	51.80%	769940.8	50.96%
-101715	6.89%	-106767	7.23%	-124352	8.42%
831173.3	50.00%	685420	41.23%	617361	37.14%
467243.5	34.17%	142213.1	10.40%	52638.67	3.85%
-55097.7	7.38%	-114171	15.30%	-164130	22.00%

C-4

Periode	Target
194	1010800
195	1314100
196	1127000
197	812500
198	896130
199	719614
200	1539750
201	1596000
202	1863667
203	1839600
204	1493784
205	566000

Test 1		Test 2		Test 3	
Error	APE	Error	APE	Error	APE
237268.8	23.47%	269997.2	26.71%	248541.1	24.59%
448327.8	34.12%	449891.6	34.24%	422247.6	32.13%
219195.2	19.45%	110331.6	9.79%	62403.25	5.54%
9759.959	1.20%	-7978.23	0.98%	-42459.5	5.23%
108300.4	12.09%	131258.7	14.65%	107663.2	12.01%
-102868	14.29%	-88640.7	12.32%	-113913	15.83%
747175.6	48.53%	779132.4	50.60%	758367.9	49.25%
207819.5	13.02%	84987.98	5.33%	52447.03	3.29%
1008334	54.10%	767572.8	41.19%	682156.9	36.60%
875533.3	47.59%	376289.2	20.45%	286362.1	15.57%
641069.9	42.92%	237998	15.93%	120877.1	8.09%
-234926	41.51%	-323703	57.19%	-386478	68.28%

206	839900
207	1492450
208	1007865
209	847721
210	850465
211	603695
212	849716
213	1607265
214	1895138
215	2133878
216	1654687
217	1116545
218	1082605

91070.8	10.84%	151124	17.99%	136946.5	16.31%
637291.8	42.70%	672066	45.03%	650180.4	43.56%
-178439	17.70%	-355417	35.26%	-400816	39.77%
62125.75	7.33%	69448.83	8.19%	40708.42	4.80%
53288.9	6.27%	71465.66	8.40%	46901.27	5.51%
-207116	34.31%	-188541	31.23%	-212830	35.25%
64434.98	7.58%	110250.6	12.97%	93729.51	11.03%
758026.3	47.16%	788482.4	49.06%	765984.8	47.66%
558535.4	29.47%	362946.7	19.15%	323619	17.08%
1152990	54.03%	630345.2	29.54%	544054.1	25.50%
663265	40.08%	-7826.03	0.47%	-80350.7	4.86%
319375.2	28.60%	199661.8	17.88%	118044.8	10.57%
296536.6	27.39%	295269.1	27.27%	263422.1	24.33%

C-6

Periode	Target	Test 1		Test 2		Test 3	
		Error	APE	Error	APE	Error	APE
219	1601811	786614.5	49.11%	770213.7	48.08%	737455	46.04%
220	900014	-189824	21.09%	-480387	53.38%	-542006	60.22%
221	894979	119472.1	13.35%	138800.6	15.51%	113254.3	12.65%
222	965381	153784.5	15.93%	166595.3	17.26%	140952	14.60%
223	621487	-202212	32.54%	-198918	32.01%	-226557	36.45%
224	1096308	315211.5	28.75%	359578.8	32.80%	342409.8	31.23%
average:			28.0854%		24.4519%		23.1304%

LAMPIRAN D
FORECAST OUTPUT

Periode	Aktual	Forecast	Abs. Error	APE
3	896000	623440	2681.818	0.43%
4	721600	640156	76648.976	13.60%
5	997000	623207	3725.8897	0.60%
6	830000	587376	68683.578	13.24%
7	1216600	600181	75983.706	14.50%
8	1108800	514922	158334.15	23.52%
9	1327428	575350	83635.07	12.69%
10	2295195	705665	143343.5	25.49%
11	1858050	654326	4374.4955	0.66%
12	1683468	601987	152858.46	20.25%
13	1261425	730988	30292.45	3.98%

Periode	Aktual	Forecast	Abs. Error	APE
14	1358602	820017	109132.48	15.35%
15	1180180	807301	179201.69	28.53%
16	1296550	729859	57461.811	7.30%
17	1235296	716878	177574.81	32.93%
18	1228282	770674	264549.9	52.27%
19	1023025	525156	203692.36	27.95%
20	1076250	584297	195697.83	25.09%
21	2021140	804602	206297.85	34.48%
22	2154850	784355	277752.94	54.83%
23	1839396	576234	1022.013	0.18%
24	1367300	524189	203910.14	28.01%
25	746200	644536	21642.466	3.47%

26	1010800	744398	100048.52	15.53%
27	1314100	651718	27513.065	4.05%
28	1127000	686181	57936.177	7.79%
29	812500	745555	80934.144	9.79%
30	896130	835343	85705.085	11.43%
31	719614	874154	93686.603	12.00%
32	520750	822663	271837.94	49.35%
33	1510812	768211	188281.92	32.47%
34	1476478	561223	68558.94	10.89%
35	1662250	606093	119546.83	16.47%
36	1367300	691717	90281.697	11.54%
37	746200	802580	209098.62	35.23%
38	1010800	784525	47545.089	5.71%

Periode	Aktual	Forecast	Abs. Error	APE
39	1314100	706385	68159.576	8.80%
40	1127000	886084	337595.99	61.55%
41	812500	761773	182271.41	31.45%
42	896130	559072	256766.29	31.47%
43	719614	686384	12807.378	1.83%
44	1539750	849658	86775.709	11.37%
45	1596000	771466	128232.39	19.94%
46	1863667	784466	54968.031	6.55%
47	1839600	753854	196855.42	35.34%
48	1493784	823997	68242.59	7.65%
49	566000	705307	17883.408	2.47%
50	839900	913535	102491.58	12.64%

51	1492450	811817	81911.14	11.22%
52	1007865	855769	254627.91	42.36%
53	847721	738041	151034.53	16.99%
54	850465	740819	20275.089	2.81%
55	603695	910702	330640.17	57.00%
56	849716	721147	188149.5	35.30%
57	1607265	569406	71548.969	11.16%
58	1895138	569630	295295.1	34.14%
59	2133878	763534	67467.029	9.69%
60	1654687	886825	79913.364	9.90%
61	1116545	786579	170212.24	17.79%
62	1082605	927577	58804.756	6.77%
63	1601811	983742	438227.99	80.33%

Periode	Aktual	Forecast	Abs. Error	APE
64	900014	845443	310744.32	58.12%
65	894979	540244	26810.892	4.73%
66	965381	542933	168639.92	23.70%
67	621487	628284	12227.648	1.91%
68	1096308	735238	46946.587	6.82%
69	516427	686245	169818.13	32.88%
70	969267	665766	303501.46	31.31%
71	945138	716986	228151.9	24.14%
72	821993	1004277	182284.28	22.18%
73	693134	968279	275145.49	39.70%
74	846338	852708	6370.4781	0.75%
75	850345	800454	49890.97	5.87%

76	936688	918857	17830.872	1.90%
77	634326	946789	312463.01	49.26%
78	825178	920293	95114.71	11.53%
79	672960	739471	66511.239	9.88%
80	658795	848859	190064.04	28.85%
81	853490	705305	148184.56	17.36%
82	758259	773858	15598.808	2.06%
83	910175	896631	13544.146	1.49%
84	960472	878082	82389.83	8.58%
85	892032	983417	91384.613	10.24%
86	829220	990126	160906.25	19.40%
87	602430	941165	338735.32	56.23%
88	547800	829692	281891.54	51.46%

Periode	Aktual	Forecast	Abs. Error	APE
89	677828	595123	82705.181	12.20%
90	526170	597168	70998.052	13.49%
91	564659	659106	94447.289	16.73%
92	692035	535107	156927.86	22.68%
93	628802	617854	10948.179	1.74%
94	849643	711957	137685.71	16.21%
95	685934	745805	59871.05	8.73%
96	695140	872324	177184.45	25.49%
97	892032	732085	159946.73	17.93%
98	829220	821621	7598.8966	0.92%
99	602430	941165	338735.32	56.23%
100	547800	829692	281891.54	51.46%

101	677828	595123	82705.181	12.20%
102	526170	597168	70998.052	13.49%
103	553380	659106	105726.29	19.11%
104	692035	531055	160980.08	23.26%
105	628802	607943	20859.255	3.32%
106	849643	711957	137685.71	16.21%
107	685934	745805	59871.05	8.73%
108	790930	872324	81394.446	10.29%
109	1272948	770956	501991.95	39.44%
110	1209825	1044266	165558.75	13.68%
111	1026675	1085446	58770.79	5.72%
112	1199120	1067703	131417.29	10.96%
113	943155	1066318	123162.71	13.06%

Periode	Aktual	Forecast	Abs. Error	APE
114	1003082	1059879	56797.23	5.66%
115	1137474	1006679	130794.92	11.50%
116	488293	1050744	562450.75	115.19%
117	1041426	997406	44019.739	4.23%
118	747409	744324	3085.093	0.41%
119	1315274	997482	317792.1	24.16%
120	1137474	1055769	81705.207	7.18%
121	728746	1082374	353628.18	48.53%
122	564245	1027267	463022.22	82.06%
123	850441	723274	127167.06	14.95%
124	768789	689995	78793.613	10.25%
125	785808	897567	111758.96	14.22%

126	637423	840595	203171.58	31.87%
127	1066476	803227	263249.45	24.68%
128	1250475	863290	387185.15	30.96%
129	1069219	1079766	10547.159	0.99%
130	1409791	1074499	335291.67	23.78%
131	1651882	1112043	539838.69	32.68%
132	1280149	1122914	157235.13	12.28%
133	873906	1090638	216731.87	24.80%
134	748426	1065569	317143.4	42.37%
135	973508	908022	65485.96	6.73%
136	791560	895793	104233.35	13.17%
137	1001392	975777	25614.919	2.56%
138	736367	933929	197561.64	26.83%

Periode	Aktual	Forecast	Abs. Error	APE
139	854281	978285	124003.97	14.52%
140	1234814	839727	395087.2	32.00%
141	964040	1044076	80036.033	8.30%
142	2024032	1065955	958076.99	47.34%
143	1741315	1483286	258029.03	14.82%
144	1713661	1094280	619380.96	36.14%
145	988692	1102509	113817.39	11.51%
146	1335362	1087152	248210.13	18.59%
147	1325262	1093563	231699.29	17.48%
148	1193555	1094437	99117.999	8.30%
149	805305	1085864	280558.75	34.84%
150	889795	1047824	158028.57	17.76%

151	807118	904036	96918.059	12.01%
152	1159590	934082	225508.09	19.45%
153	1119370	1000784	118586.5	10.59%
154	1894362	1070645	823716.85	43.48%
155	1239350	1290066	50715.712	4.09%
156	812800	1089487	276686.73	34.04%
157	657500	1056332	398832.04	60.66%
158	820700	833542	12841.543	1.56%
159	896000	758182	137817.73	15.38%
160	721600	916290	194690.01	26.98%
161	997000	915615	81384.96	8.16%
162	830000	887181	57180.57	6.89%
163	1216600	993517	223082.83	18.34%

Periode	Aktual	Forecast	Abs. Error	APE
164	1108800	1030747	78053.027	7.04%
165	1327428	1074758	252669.59	19.03%
166	2295195	1094910	1200285.1	52.30%
167	1858050	1426164	431885.8	23.24%
168	1683468	1091932	591536.41	35.14%
169	1261425	1096814	164610.9	13.05%
170	1358602	1090202	268400.48	19.76%
171	1180180	1098297	81882.666	6.94%
172	1296550	1085968	210581.63	16.24%
173	1235296	1091451	143844.61	11.64%
174	1228282	1087857	140424.94	11.43%
175	1023025	1085724	62699.316	6.13%

176	1076250	1069352	6897.809	0.64%
177	2021140	1043892	977247.96	48.35%
178	2154850	1402932	751917.51	34.89%
179	1839396	1113989	725407.2	39.44%
180	1367300	1093705	273594.76	20.01%
181	746200	1090850	344649.63	46.19%
182	1010800	1068663	57862.596	5.72%
183	1314100	909485	404614.92	30.79%
184	1127000	1089195	37804.793	3.35%
185	812500	1081750	269249.74	33.14%
186	896130	1033729	137598.88	15.35%
187	719614	910956	191342.07	26.59%
188	520750	915178	394428.48	75.74%

Periode	Aktual	Forecast	Abs. Error	APE
189	1510812	698286	812525.79	53.78%
190	1476478	1182218	294259.73	19.93%
191	1662250	1099716	562534.34	33.84%
192	1367300	1117162	250137.81	18.29%
193	746200	1092394	346193.66	46.39%
194	1010800	1068663	57862.596	5.72%
195	1314100	909485	404614.92	30.79%
196	1127000	1089195	37804.793	3.35%
197	812500	1081750	269249.74	33.14%
198	896130	1033729	137598.88	15.35%
199	719614	910956	191342.07	26.59%
200	1539750	915178	624571.52	40.56%

201	1596000	1208432	387567.55	24.28%
202	1863667	1106117	757550.49	40.65%
203	1839600	1129012	710587.71	38.63%
204	1493784	1103007	390776.78	26.16%
205	566000	1092683	526683.47	93.05%
206	839900	1073444	233543.99	27.81%
207	1492450	686280	806169.74	54.02%
208	1007865	1161143	153277.74	15.21%
209	847721	1083703	235982.2	27.84%
210	850465	1000984	150519.47	17.70%
211	603695	919785	316090.14	52.36%
212	849716	847982	1733.8422	0.20%
213	1607265	723783	883482.29	54.97%

Periode	Aktual	Forecast	Abs. Error	APE
214	1895138	1229944	665194.1	35.10%
215	2133878	1132102	1001776.2	46.95%
216	1654687	1125443	529243.64	31.98%
217	1116545	1091423	25122.125	2.25%
218	1082605	1088059	5453.8682	0.50%
219	1601811	1061466	540345.44	33.73%
220	900014	1165960	265946.44	29.55%
221	894979	1084518	189538.76	21.18%
222	965381	962315	3066.0748	0.32%
223	621487	977697	356210.11	57.32%
224	1096308	935364	160943.98	14.68%
average :				22.49%

